LSI-11

LSI-11 FIS INSTR. TESTS

MD-11-DVKAC-B

EP-DVKAC-B-DL-A

FICHE 1 OF 1

APR 1977

digital

MADE IN USA

IDENTIFICATION SEQ 0001

PRODUCT CODE: MAINDEC-11-DVKAC-B-D

PRODUCT NAME: LSI-11 FIS INSTRUCTION TESTS

DATE: MARCH,1977

MAINTAINER: DIAGNOSTIC GROUP

AUTHOR: PERVEZ A. ZAKI

MODIFIED BY: ALAN W. BOSTICK

DATE MODIFIED: JANUARY 1977

CONTENTS

1. ABSTRACT

SEQ 0003

THIS PROGRAM TESTS THE LSI-11 FLOATING INSTRUCTION SET (FADD, FSUB, FMUL, AND FDIV) OPTION WITH FIXED NUMBER PATTERNS, USING EACH REGISTER AT LEAST ONCE AS THE STACK POINTER. IT ALSO CHECKS STACK OVERFLOW AND THAT THE FLOATING INSTRUCTIONS CAN BE INTERRUPTED (BY THE CONSOLE TELETYPE) (HOWEVER THIS TEST WILL NOT BE EXECUTED WHEN BIT 5 OF $ENVM BYTE IS HIGH). THE PROGRAM SHOULD BE RUN FOR AT LEAST 2 PASSES WITH ALL SWITCHES LOW. THE PROGRAM IS DESIGNED TO RUN UNDER APT. AND ACT. SYSTEMS. WHEN RUNNING UNDER APT. WITH BIT 5 OF $ENVM LOW IT WILL BE REQUIRED TO HAVE A SLU WITH TTY REGISTERS HAVING ADDRESSES OF 176560-66 AND INTERUPT VECTORS OF 70 FOR RECEIVER AND 74 FOR TRANSMIT-TER. UNDER SUCH A CONDITION IT WILL ALSO BE REQUIRED TO CHANGE THE RUN TIME OF FIRST PASS FROM 5 SECONDS TO THE TIME GIVEN IN SEC. 8.1, AND THE RUN TIME FOR THE LONGEST TEST FROM 3 SECONDS TO 30 SECONDS.

2. REQUIREMENTS

2.1 EQUIPMENT

LSI-11 STANDARD COMPUTER WITH FIS OPTION
AND 4K OF MEMORY

2.2 STORAGE

PROGRAM STORAGE - THE ROUTINES USE MEMORY 0 - 17500

2.3 PRELIMINARY PROGRAMS

NONE

3. LOADING PROCEDURE

USE STANDARD PROCEDURE FOR ABS TAPES.

4. STARTING PROCEDURE

4.1 CONTROL SWITCH SETTINGS

SEE 5.1.1 (ALL LOW FOR WORST CASE TESTING)

4.2 STARTING ADDRESS

AFTER LOADING THE PROGRAM IT SHOULD ALWAYS BE STARTED AT 200. IF IT IS DESIRED TO SAVE THE PASS COUNTER THEN THE PROGRAM SHOULD BE RESTARTED AT LOCATION RESTRT [I.E. 222] OTHERWISE THE PROGRAM CAN BE RESTARTED AT 200

4.3 PROGRAM AND/OR OPERATOR ACTION

4.3.1 STAND ALONE

1) LOAD PROGRAM INTO MEMORY USING ABS LOADER.
2) SET SWITCHES (SEE SEC 5.1.1) ALL LOW EXCEPT BIT 7 FOR WORST CASE.
3) TYPE 200G
4) THE PROGRAM WILL LOOP AND "END PASS" WILL BE TYPED AFTER COMPLETION OF EVERY PASS. HOWEVER TYPE OUT WILL BE SUPPRESSED IF BIT 5 OF LOCATION $ENVM IS HIGH
5) A MINIMUM OF TWO PASSES SHOULD ALWAYS BE RUN.

4.3.2 UNDER APT

LOAD THE PROGRAM, SET THE SWITHES (SEE SEC. 5.1.1) AND START. WHEN UNDER APT. WITH BIT 5 OF $ENVM LOW IT WILL BE REQUIRED TO HAVE A SLU WITH TTY REGISTERS HAVING ADDRESSES OF 176560-66 AND INTERUPT VECTORS OF 70 FOR RECEIVER AND 74 FOR TRANSMIT-TER. UNDER SUCH A CONDITION IT WILL ALSO BE REQUIRED TO CHANGE THE RUN TIME OF FIRST PASS FROM 5 SECONDS TO THE TIME GIVEN IN SEC. 8.1, AND THE RUN TIME FOR THE LONGEST TEST FROM 3 SECONDS TO 30 SECONDS. THE TEST TIMES AND PASS TIMES ARE SUGGESTED WITH BIT 7 OF $SWREG. LOW, IF IT IS DESIRED TO ENABLE THE ITERATIONS THEN THE TIMES SHOULD BE MULTIPLIED BY A FACTOR OF 256.

5. OPERATING PROCEDURE

5.1 OPERATIONAL SWITCH SETTINGS

ALL SWITCHES LOW EXCEPT SW<11> IS WORST CASE TESTING. WITH BIT 11 OF THE LOCATION $SWREG [I.E. LOCATION 422], HIGH EACH SUBTEST WILL BE LOOPED UPON UNTIL COMPLETION OF 256 PASSES OF THAT SUBTEST. "END PASS" WILL BE TYPED UPON COMPLETION OF A PASS OF THE ENTIRE PROGRAM. ALTERNATE PASS WILL RUN WITH THE T-BIT SET.

5.1.1 SWITCH SETTINGS ARE:

A 16 BIT LOCATION CALLED $SWREG (I.E. LOCATION 422) HAS BEEN USED TO GIVE THE FOLLOWING OPTIONS BY INSERTING A 1 IN THEIR RESPECTIVE POSITIONS

F01

BIT #	OCTAL VALUE	FUNCTION
15	100000.....	HALT ON ERROR
14	040000.....	SCOPE LOOP
13	020000.....	INHIBIT PRINTOUT
12	010000.....	INHIBIT TRACE TRAPPING
11	004000.....	ENABLE ITERATIONS OF SUBTEST
10	002000.....	BELL ON ERROR
09	001000.....	LOOP ON ERROR
08	0004XX.....	LOOP ON TEST IN BITS 7 THRU 0

AN 8 BIT BYTE $ENVM [I.E. LOCATION 421] HAS BEEN USED TO DEFINE THE OPERATING MODE. ALL TYPEOUTS CAN BE SUPPRESSED BY MAKING BIT 5 OF BYTE $ENVM HIGH, IN OTHER WORDS BY PLCING A 20000 IN LOCATION 420

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$". IF A SCOPE LOOP IS REQUESTED, THE CURRENT SUBTEST WILL BE LOOPED UPON. SW<11> IS A 1 INHIBITS ITERATION OF SUBTESTS. THE CONTENTS OF "LAD$" 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 A 1 AND A HLT IS EXECUTED, THE SUBTEST WILL BE LOOPED UPON UNTIL 256 CONSECUTIVE GOOD PASSES ARE COMPLETED. TO INHIBIT TYPEOUTS, MAKE SW<13> A 1. TO RING THE BELL ON AN ERROR, MAKE SW<10> A 1. A HIGH BIT 5 IN LOCATION $ENVM WILL INHIBIT ANY TYPEOUTS AND RINGING OF BELLS

5.2.3 T BIT TRAP

IF SW<12> IS 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.

5.2.4 TRAP CATCHER

A ".+2" - "HALT" SEQUENCE IS REPEATED FROM 0-776 TO CATCH ANY UNEXPECTED TRAPS. THUS ANY UNEXPECTED TRAPS OR INTERRUPTS WILL HALT AT THE VECTOR +2.

5.2.5 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 OCCURRED WHICH CHECKS THE DATA ON THE STACK(S). IF AN ERROR WAS NOT ANTICIPATED, AN ERRONEOUS TRAP WILL BE DETECTED IN TRAPER.

5.2.6 NOP

A NOP IS PLACED JUST BEFORE EACH FIS INSTRUCTION. THIS ALLOWS THE OPERATOR TO PATCH IN A HALT FOR DEBUGGING PURPOSES

6. ERRORS

6.1 ERROR PRINTOUT

THE FORMAT IS AS FOLLOWS:

ERRNM ADR PS SP ANS1 ANS2 ANS3 ANS4 ANS5 ANS6

WHERE:

ERRNM = ERROR NUMBER
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 TO 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 ERRNM, ADR, PS, AND SP.

TO FIND THE FAILING TEST, LOOP AT THE LISTING ABOVE THE ADDRESS TYPED. IN MOST CASES THE COMMENT BESIDE THE HLT TELLS WHAT WAS BEING CHECKED AND WHAT WAS EXPECTED. ALL PRINTOUTS WILL BE SUPPRESSED WHEN BIT 5 OF LOCATION $ENVM IS HIGH. WHILE RUNNING UNDER APT THE DIAGNOSTIC WILL NOT SUPPORT SPOOLING OF CONSOLE OUTPUTS.

6.2 ERROR RECOVERY

RESTART AT 200 OR 222 (SEE SEC 4.2)

6.3 ERROR COUNTER

AN ERROR COUNT IS KEPT IN LOCATION "ERRORS". 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 INTERRUPT TESTS, THE EXECUTION TIME CAN BE HALF A MINUTE OR MORE. HOWEVER, NORMALLY "END PASS" WILL BE TYPED WITHIN 40 SECONDS WITH ALL SWITCHES DOWN. EXECUTION TIME WILL INCREASE BY A FACTOR OF 256 WHEN ITTERATIONS OF EACH SUBTEST ARE ENABLED.

8.2 STACK POINTER

STACK IS INITIALLY SET TO 600

8.3 PASS COUNT

A 16 BIT LOCATION "$PASS" (I.E. LOCATION 406) IS USED TO KEEP PASS COUNT. IT CAN BE CLEARED BY RESTARTING THE PROGRAM AT 200

8.4 TEST NUMBER

A 16 BIT LOCATION "$TESTN" (I.E. LOCATION 404) IS USED TO KEEP TRACK OF THE TEST NUMBER, UPPER BYTE OF THIS LOCATION GIVES THE ITERATION NUMBER AND THE LOWER BYTE THE TEST THAT WAS BEING EXECUTED

8.5 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 INTERRUPTED WITH NO OTHER ERROR TYPEOUTS.

9. PROGRAM DESCRIPTION

THIS PROGRAM TESTS ALL THE FIS INSTRUCTIONS OF THE LSI-11 (FADD, FSUB, FMUL, AND FDIV). ALL REGISTERS ARE CHECKED TO SEE

IF THEY FUNCTION PROPERLY AS THE STACK POINTER. THE PROGRAM
HAS MANY SUBTESTS (THE CODE BETWEEN 2 SCOPE STATEMENTS)
WHICH ARE RUN ONCE BEFORE CONTINUING TO THE NEXT SUBTEST.
SW<11> SET TO A 1 CAUSES EACH SUBTEST TO BE RUN 256 TIMES
SW<9> SET TO A 1 ENABLES LOOP ON ERROR. THE LOCATION $ICNT
CONTAINS THE ITERATION COUNT AND THE LOCATION $TESTN
CONTAINS THE TEST NUMBER. 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
LOCATION $SWREG AND SW<8> SET TO A 1. THIS TEST WILL BE LOOPED
UPON UNTIL SW<8> IS SET TO A 0 OR THE RIGHT BYTE IS CHANGED.
IF THE TEST IS NON-EXISTENT, THE PROGRAM WILL BE RUN AS
USUAL.

DVKACB.P11 13-JAN-77 15:28 TABLE OF CONTENTS

```
1
```

```
                                .ABS
                                .TITLE  DVKACB
                                ;*COPYRIGHT (C) AUGUST 1975
                                ;*DIGITAL EQUIPMENT CORP.
                                ;*MAYNARD, MASS. 01754
                                ;*
                                ;*PROGRAM BY PERVEZ ZAKI
                                ;*
                                ;*THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC
                                ;*PACKAGE (MAINDEC-11-DZQAC-B),JULY 11,1975.
                                ;*
000001                          $TN=1
160000                          $SWR=160000     ;;HALT ON ERROR, LOOP ON TEST, INHIBIT ERROR TYPOUT

                        ;       SWITCH                  USE
                        ;       ------          --------------------
                        ;         8             LOOP ON TEST IN SW<7:0>
                        ;         9             LOOP ON ERROR
                        ;        10             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:
                        ;  ERRNM   ADR  PSW  SP  ANS1 ANS2 ANS3 ANS4 ANS5 ANS6
                        ;
                        ;WHERE  ERRNM=  ERROR NUMBER
                        ;       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.

104000                  HLT=    EMT
000000                  R0=     %0
```

```
                        R1=     %1
                        R2=     %2
                        R3=     %3
                        R4=     %4
                        R5=     %5
                        TTY=    %5
                        SP=     %6
                        PC=     %7
                        PWRVEC= 24
                        SCOPE=  TRAP
                        SW15=   100000
                        SW14=    40000
                        SW13=    20000
                        SW12=    10000
                        SW11=     4000
                        SW10=     2000
                        SW09=     1000
                        SW08=      400
                        TYPE=   IOT
                        N=      1
                        $F=     1
                        ;*****************************************************************

000000                  .=      0                       ;TRAP CATCHER FROM 0 - 776

                        ;*****************************************************************

                        .SBTTL  ACT11 HOOKS
                        ;HOOKS REQUIRED BY ACT11
001000                          $SVPC=.                 ;SAVE PC
000046                          .=46
000046  015650                  $ENDAD                  ;;1)SET LOC.46 TO ADDRESS OF $ENDAD IN .$EOP
000052                          .=52
000052  000000                  .WORD   0               ;;2)SET LOC.52 TO ZERO
001000                          .=$SVPC                 ;; RESTORE PC
```

```
                        000400                  .=      400
                                                ;*****************************************************************

                                                .SBTTL  APT MAILBOX-ETABLE


                                                        .EVEN
000400                                          $MAIL:                  ;;APT MAILBOX
000400  000000                                  $MSGTY: .WORD   AMSGTY   ;;MESSAGE TYPE CODE
000402  000000                                  $FATAL: .WORD   AFATAL   ;;FATAL ERROR NUMBER
000404  000000                                  $TESTN: .WORD   ATESTN   ;;TEST NUMBER
000406  000000                                  $PASS:  .WORD   APASS    ;;PASS COUNT
000410  000000                                  $DEVCT: .WORD   ADEVCT   ;;DEVICE COUNT
000412  000000                                  $UNIT:  .WORD   AUNIT    ;;I/O UNIT NUMBER
000414  000000                                  $MSGAD: .WORD   AMSGAD   ;;MESSAGE ADDRESS
000416  000000                                  $MSGLG: .WORD   AMSGLG   ;;MESSAGE LENGTH
000420                                          $ETABLE:                ;;APT ENVIRONMENT TABLE
000420     000                                  $ENV:   .BYTE   AENV     ;;ENVIRONMENT BYTE
000421     000                                  $ENVM:  .BYTE   AENVM    ;;ENVIRONMENT MODE BITS
000422  000000                                  $SWREG: .WORD   ASWREG   ;;APT SWITCH REGISTER
000424  000000                                  $USWR:  .WORD   AUSWR    ;;USER SWITCHES
000426  000000                                  $CPUOP: .WORD   ACPUOP   ;;CPU TYPE,OPTIONS
                                                ;*                      ;;BITS 15-11=CPU TYPE
                                                ;*                      ;;      11/04=01,11/05=02,11/20=03,11/40=04,11/45=05
                                                ;*                      ;;      11/70=06,PDQ=07,Q=10
                                                ;*                      ;;BIT 10=REAL TIME CLOCK
                                                ;*                      ;;BIT  9=FLOATING POINT PROCESSOR
                                                ;*                      ;;BIT  8=MEMORY MANAGEMENT
000430                                          $ETEND:
                                                        .MEXIT
                                                ;*****************************************************************

                                                .SBTTL  APT PARAMETER BLOCK
                                                ;SET LOCATIONS 24 AND 44 AS REQUIRED FOR APT
                                                ;*****************************************************************
                        000430                          .$X=.   ;;SAVE CURRENT LOCATION
                        000024                          .=24    ;;SET POWER FAIL TO POINT TO START OF PROGRAM
000024  000200                                         200     ;;FOR APT START UP
                        000044                          .=44    ;;POINT TO APT INDIRECT ADDRESS PNTR.
000044  000430                                         $APTHDR ;;POINT TO APT HEADER BLOCK
                        000430                          .=.$X   ;;RESET LOCATION COUNTER
                                                ;*****************************************************************
                                                ;SETUP APT PARAMETER BLOCK AS DEFINED IN THE APT-PDP11 DIAGNOSTIC
                                                ;INTERFACE SPEC.

000430                                          $APTHD:
000430  000000                                  $HIBTS: .WORD   0               ;;TWO HIGH BITS OF 18 BIT MAILBOX ADDR.
000432  000400                                  $MBADR: .WORD   $MAIL           ;;ADDRESS OF APT MAILBOX (BITS 0-15)
000434  000003                                  $TSTM:  .WORD   3               ;;RUN TIM OF LONGEST TEST
000436  000005                                  $PASTM: .WORD   5               ;;RUN TIME IN SECS. OF 1ST PASS ON 1 UNIT (QUICK VERIFY)
000440  000000                                  $UNITM: .WORD                   ;;ADDITIONAL RUN TIME (SECS) OF A PASS FOR EACH ADDITIONAL UNIT
000442  000014                                          .WORD   $ETEND-$MAIL/2 ;;LENGTH MAILBOX-ETABLE(WORDS)
                        000430                          .=      $APTHD
000430                                          HLTAD$:
```

```
                    000432                          .=      HLTAD$+2
000432              000434      $PSW:               .=      $PSW+2          ;PROCESSOR STATUS WORD
000434              000436      $SP:                .=      $SP+2           ;STACK POINTER
000436              000440      ANS1:               .=      ANS1+2          ;FIRST ANSWER (SEE CODE)
000440              000442      ANS2:               .=      ANS2+2
000442              000444      ANS3:               .=      ANS3+2
000444  000000                  ANS4:   0
000446  000000                  ANS5:   0
000450  000000                  ANS6:   0
000452  000000  000000  000000          0,0,0,0                             ;NON-%6 STACK BUFFER
000460  000000
000462  000000                  ERRORS: 0
000464  000244                  FISVEC: 244                                 ;FIS TRAP VECTOR ADDRESS
000466  000246                  FISLVL: 246
000470  000000                  LAD$:   0
000472  000412      000         RETURN: .ASCIZ  <12><15>
000475      015     020012  020040  SPACE:  .ASCIZ  <15><12>"   "           ;RETURN AND 3 SPACES
000502      000
000503      000                 $ICNT:  .BYTE   0
                                        .EVEN
000504  000007                  $BELL:  .WORD   7                           ;RING A BELL
000506  000000                  SAVTPS: 0                                   ;LOC TO SAVE TELEPRINTER STATUS
000510  000000                  STACK0: 0                                   ;NON-%6 STACK NORMAL LIMIT
000512  000000                  STACK2: 0                                   ;
000514  000000                  STACK4: 0                                   ;
000516  000000                  STACK6: 0                                   ;
000520  000000  000000  000000  STACK8: 0,0,0,0,0                         ;NON-%6 STACK BUFFER
000526  000000  000000
000532  000000                  STAK10: 0
                    000511      STACK1 = STACK0+1
000534  000000                  TEMP:   0
000536  000000                  TIMES:  0
000540  000000                  TYPCNT: 0
000542  000006                  YESRT:  RTT                                 ;RETURN FROM TRACE TRAP
000544  000000                  .PR:    0                                   ;COUNT AND SWITCH
000546  000064                  TTYOUT: 64
000550  177564                  $TPS:   177564                              ;TTY PRINTER STATUS REG.
000552  177566                  $TPB:   177566                              ;TTY PRINTER BUFFER REG.
```

```
                        ;*****************************************************************

                        .SBTTL  STARTING OF THE PROGRAM


                000200  .=      200
000200  012767  000001  000330  MOV     #1,TIMES        ;NUMBER OF ITERATIONS IN THE FIRST PASS=1
000206  012700  000410          MOV     #$DEVCT,R0      ;PREPARE TO INITIALIZE THE PROGRAM
000212  005040          2$:     CLR     -(R0)
000214  022700  000400          CMP     #$MAIL,R0
000220  001374                  BNE     2$

000222  000167  000352  RESTRT: JMP     BEGIN           ;JUMP TO STARTING ADDRESS OF PROGRAM


                000600  .=      600

000600  012706  000600  BEGIN:  MOV     #BEGIN, SP      ;INITIALIZE STACK POINTER
000604  012737  000542  000014  MOV     #YESRT, @#14    ;SET TRACE TRAP VECTOR
000612  012737  017154  000020  MOV     #$TYPE, @#20    ;SET UP VECTOR 20
000620  012737  017014  000024  MOV     #$PWRDN ,@#24   ;SERVICE POWER DOWN ROUTINE FOR ANY FUTURE
                                                        ;POWER DOWN
000626  012700  000030          MOV     #30,    R0      ;SET R0 TO VECTOR 30
000632  012720  016450          MOV     #HLT$,  (0)+    ;SET EMT VECTOR
000636  012720  000340          MOV     #340,   (0)+
000642  012720  015702          MOV     #SCOPE$,(0)+    ;SET TRAP VECTOR
000646  012710  000340          MOV     #340,   (0)
000652  012737  000006  000004  1$:     MOV     #6,     @#4     ;RESTORE TIME-OUT VECTOR
000660  132737  000001  000420  BITB    #1,@#SENV       ;ARE WE UNDER APT ?
000666  001410                  BEQ     2$              ;IF NOT THEN GO TO 2$
000670  012700  000554          MOV     #$TPB+2,R0      ;OTHERWISE SET FOR THE OTHER SLU
000674  012740  176566          MOV     #176566,-(R0)
000700  012740  176564          MOV     #176564,-(R0)
000704  012740  000074          MOV     #74,-(R0)
000710  005067  177470  2$:     CLR     STE$TN
000714  005067  177550          CLR     LAD$            ;CLEAR LOOP ADDRESS
```

```
                                ;*************************************************************************
                                ;TEST 1:        FADD (LSI-11 FLOATING ADD INSTRUCTION)
                                ;       000000,000000 + 000000,000000 = 000000,000000
                                ;       PS = 004,       STACK POINTER = R0
                                ;*************************************************************************

000720  104400                          SCOPE
000722  004567  015244          TST1:   JSR     R5,     PUSHR   ;PUSH 4 WORDS ONTO R0 STACK, SET PRIORITY
000726  000000  000000                  .WORD   000000,000000   ;SECOND OPERAND ON TOP
000732  000000  000000                  .WORD   000000,000000   ;FIRST OPERAND ON BOTTOM
000736  000000                          .WORD   000             ;PROCESSOR PRIORITY LEVEL
000740  016442  000340                  .WORD   TRAPER,340      ;FIS TRAP VECTOR
000744  012700  000510                  MOV     #STACK0,R0      ;SET UP STACK POINTER

000750  000240                          NOP
000752  075000                          FADD    R0              ;FLOATING ADD ON THE R0 STACK

000754  004767  015244                  JSR     PC,     POPR    ;POP THE ANSWER
000760  010067  177450                  MOV     R0,     $SP     ;SAVE "STACK POINTER"
000764  122767  000004  177440          CMPB    #004,   $PSW    ;CHECK PS (EXCEPT T BIT)
000772  001402                          BEQ     .+6             ;BRANCH IF OK
000774  104000                          HLT                     ;PS NOT EQUAL TO 004
000776  000001                          1                       ;THE ERROR NUMBER IS 1

001000  022767  000514  177426          CMP     #STACK4,$SP     ;CHECK THE STACK POINTER (R0)
001006  001402                          BEQ     .+6             ;BRANCH IF OK
001010  104000                          HLT                     ;STACK POINTER (R0) NOT EQUAL TO #STACK4
001012  000002                          2                       ;THE ERROR NUMBER IS 2

001014  005767  177416                  TST     ANS1            ;CHECK FIRST HALF OF ANSWER
001020  001402                          BEQ     .+6             ;BRANCH IF OK
001022  104002                          HLT+2                   ;ANS1 NOT EQUAL TO 000000
001024  000003                          3                       ;THE ERROR NUMBER IS 3

001026  005767  177406                  TST     ANS2            ;CHECK SECOND HALF OF ANSWER
001032  001402                          BEQ     .+6             ;BRANCH IF OK
001034  104002                          HLT+2                   ;ANS2 NOT EQUAL TO 000000
001036  000004                          4                       ;THE ERROR NUMBER IS 4

001040  122767  000001  177336  END1:   CMPB    #1,     $TESTN  ;CHECK THE TEST NUMBER
001046  001402                          BEQ     .+6             ;BRANCH IF OK
001050  104000                          HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
001052  000005                          5                       ;THE ERROR NUMBER IS 5
```

```
                        ;*************************************************************************
                        ;TEST 2:        FADD (LSI-11 FLOATING ADD INSTRUCTION)
                        ;       000000,000000 + 152525,052524 = 152525,052524
                        ;       PS = 010,       STACK POINTER = R3
                        ;*************************************************************************

001054  104400                  SCOPE
001056  004567  015110  TST2:   JSR     R5,     PUSHR   ;PUSH 4 WORDS ONTO R3 STACK, SET PRIORITY
001062  152525  052524          .WORD   152525,052524   ;SECOND OPERAND ON TOP
001066  000000  000000          .WORD   000000,000000   ;FIRST OPERAND ON BOTTOM
001072  000040                  .WORD   040             ;PROCESSOR PRIORITY LEVEL
001074  016442  000340          .WORD   TRAPER,340      ;FIS TRAP VECTOR
001100  012703  000510          MOV     #STACK0,R3      ;SET UP STACK POINTER

001104  000240                  NOP
001106  075003                  FADD    R3              ;FLOATING ADD ON THE R3 STACK

001110  004767  015110          JSR     PC,     POPR    ;POP THE ANSWER
001114  010367  177314          MOV     R3,     $SP     ;SAVE "STACK POINTER"
001120  122767  000010  177304  CMPB    #010,   $PSW    ;CHECK PS (EXCEPT T BIT)
001126  001402                  BEQ     .+6             ;BRANCH IF OK
001130  104000                  HLT                     ;PS NOT EQUAL TO 010
001132  000006                  6                       ;THE ERROR NUMBER IS 6

001134  022767  000514  177272  CMP     #STACK4,$SP     ;CHECK THE STACK POINTER (R3)
001142  001402                  BEQ     .+6             ;BRANCH IF OK
001144  104000                  HLT                     ;STACK POINTER (R3) NOT EQUAL TO #STACK4
001146  000007                  7                       ;THE ERROR NUMBER IS 7

001150  022767  152525  177260  CMP     #152525,ANS1    ;CHECK FIRST HALF OF ANSWER
001156  001402                  BEQ     .+6             ;BRANCH IF OK
001160  104002                  HLT+2                   ;ANS1 NOT EQUAL TO 152525
001162  000010                  10                      ;THE ERROR NUMBER IS 10

001164  022767  052524  177246  CMP     #052524,ANS2    ;CHECK SECOND HALF OF ANSWER
001172  001402                  BEQ     .+6             ;BRANCH IF OK
001174  104002                  HLT+2                   ;ANS2 NOT EQUAL TO 052524
001176  000011                  11                      ;THE ERROR NUMBER IS 11

001200  122767  000002  177176  END2:   CMPB    #2,     $TESTN  ;CHECK THE TEST NUMBER
001206  001402                  BEQ     .+6             ;BRANCH IF OK
001210  104000                  HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
001212  000012                  12                      ;THE ERROR NUMBER IS 12
```

```
                                ;*************************************************************************
                                ;TEST 3:        FADD (LSI-11 FLOATING ADD INSTRUCTION)
                                ;       040200,000000 + 040200,000000 = 040400,000000
                                ;       PS = 000,       STACK POINTER = R1
                                ;*************************************************************************

001214  104400                          SCOPE
001216  004567  014750          TST3:   JSR     R5,     PUSHR   ;PUSH 4 WORDS ONTO R1 STACK, SET PRIORITY
001222  040200  000000                  .WORD   040200,000000   ;SECOND OPERAND ON TOP
001226  040200  000000                  .WORD   040200,000000   ;FIRST OPERAND ON BOTTOM
001232  000040                          .WORD   040             ;PROCESSOR PRIORITY LEVEL
001234  016442  000340                  .WORD   TRAPER,340      ;FIS TRAP VECTOR
001240  012701  000510                  MOV     #STACK0,R1      ;SET UP STACK POINTER

001244  000240                          NOP
001246  075001                          FADD    R1              ;FLOATING ADD ON THE R1 STACK

001250  004767  014750                  JSR     PC,     POPR    ;POP THE ANSWER
001254  010167  177154                  MOV     R1,     $SP     ;SAVE "STACK POINTER"
001260  105767  177146                  TSTB    $PSW            ;CHECK PS (EXCEPT T BIT)
001264  001402                          BEQ     .+6             ;BRANCH IF OK
001266  104000                          HLT                     ;PS NOT EQUAL TO 000
001270  000013                          13                      ;THE ERROR NUMBER IS 13

001272  022767  000514  177134          CMP     #STACK4,$SP     ;CHECK THE STACK POINTER (R1)
001300  001402                          BEQ     .+6             ;BRANCH IF OK
001302  104000                          HLT                     ;STACK POINTER (R1) NOT EQUAL TO #STACK4
001304  000014                          14                      ;THE ERROR NUMBER IS 14

001306  022767  040400  177122          CMP     #040400,ANS1    ;CHECK FIRST HALF OF ANSWER
001314  001402                          BEQ     .+6             ;BRANCH IF OK
001316  104002                          HLT+2                   ;ANS1 NOT EQUAL TO 040400
001320  000015                          15                      ;THE ERROR NUMBER IS 15

001322  005767  177112                  TST     ANS2            ;CHECK SECOND HALF OF ANSWER
001326  001402                          BEQ     .+6             ;BRANCH IF OK
001330  104002                          HLT+2                   ;ANS2 NOT EQUAL TO 000000
001332  000016                          16                      ;THE ERROR NUMBER IS 16

001334  122767  000003  177042  END3:   CMPB    #3,     $TESTN  ;CHECK THE TEST NUMBER
001342  001402                          BEQ     .+6             ;BRANCH IF OK
001344  104000                          HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
001346  000017                          17                      ;THE ERROR NUMBER IS 17
```

```
                        ;*************************************************************************
                        ;TEST 4:        FADD (LSI-11 FLOATING ADD INSTRUCTION)
                        ;       077777,177777 + 177777,177777 = 000000,000000
                        ;       PS = 004,       STACK POINTER = R2
                        ;*************************************************************************

001350  104400                  SCOPE
001352  004567  014614  TST4:   JSR     R5,     PUSHR   ;PUSH 4 WORDS ONTO R2 STACK, SET PRIORITY
001356  177777  177777          .WORD   177777,177777   ;SECOND OPERAND ON TOP
001362  077777  177777          .WORD   077777,177777   ;FIRST OPERAND ON BOTTOM
001366  000100                  .WORD   100             ;PROCESSOR PRIORITY LEVEL
001370  016442  000340          .WORD   TRAPER,340      ;FIS TRAP VECTOR
001374  012702  000510          MOV     #STACK0,R2      ;SET UP STACK POINTER

001400  000240                  NOP
001402  075002                  FADD    R2              ;FLOATING ADD ON THE R2 STACK

001404  004767  014614          JSR     PC,     POPR    ;POP THE ANSWER
001410  010267  177020          MOV     R2,     $SP     ;SAVE "STACK POINTER"
001414  122767  000004  177010  CMPB    #004,   $PSW    ;CHECK PS (EXCEPT T BIT)
001422  001402                  BEQ     .+6             ;BRANCH IF OK
001424  104000                  HLT                     ;PS NOT EQUAL TO 004
001426  000020                  20                      ;THE ERROR NUMBER IS 20

001430  022767  000514  176776  CMP     #STACK4,$SP     ;CHECK THE STACK POINTER (R2)
001436  001402                  BEQ     .+6             ;BRANCH IF OK
001440  104000                  HLT                     ;STACK POINTER (R2) NOT EQUAL TO #STACK4
001442  000021                  21                      ;THE ERROR NUMBER IS 21

001444  005767  176766          TST     ANS1            ;CHECK FIRST HALF OF ANSWER
001450  001402                  BEQ     .+6             ;BRANCH IF OK
001452  104002                  HLT+2                   ;ANS1 NOT EQUAL TO 000000
001454  000022                  22                      ;THE ERROR NUMBER IS 22

001456  005767  176756          TST     ANS2            ;CHECK SECOND HALF OF ANSWER
001462  001402                  BEQ     .+6             ;BRANCH IF OK
001464  104002                  HLT+2                   ;ANS2 NOT EQUAL TO 000000
001466  000023                  23                      ;THE ERROR NUMBER IS 23

001470  122767  000004  176706  END4:   CMPB    #4,     $TESTN  ;CHECK THE TEST NUMBER
001476  001402                  BEQ     .+6             ;BRANCH IF OK
001500  104000                  HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
001502  000024                  24                      ;THE ERROR NUMBER IS 24
```

```
                        ;*************************************************************************
                        ;TEST 5:        FADD (LSI-11 FLOATING ADD INSTRUCTION)
                        ;       052525,052525 + 152525,052524 = 044600,000000
                        ;       PS = 200,       STACK POINTER = SP
                        ;*************************************************************************

001504  104400                  SCOPE
001506  004567  014306  TST5:   JSR     R5,     PUSHS   ;PUSH 4 WORDS ONTO STACK, SET PRIORITY
001512  152525  052524          .WORD   152525,052524   ;SECOND OPERAND ON TOP
001516  052525  052525          .WORD   052525,052525   ;FIRST OPERAND ON BOTTOM
001522  000217                  .WORD   217             ;PROCESSOR PRIORITY LEVEL
001524  016442  000340          .WORD   TRAPER,340      ;FIS TRAP VECTOR

001530  000240                  NOP
001532  075006                  FADD    SP              ;FLOATING ADD ON THE STACK

001534  004767  014320          JSR     PC,     POPS    ;POP THE ANSWER
001540  022706  000600          CMP     #BEGIN, SP      ;CHECK THE STACK POINTER
001544  001405                  BEQ     TSA5            ;BRANCH IF OK
001546  012706  000600          MOV     #BEGIN, SP      ;RESTORE STACK POINTER
001552  104000                  HLT                     ;STACK POINTER FOULED UP
001554  000025                  25                      ;THE ERROR NUMBER IS 25
001556  000421                  BR      ENDS            ;SKIP REST OF TEST

001560  122767  000200  176644  TSA5:   CMPB    #200,   $PSW    ;CHECK PS (EXCEPT T BIT)
001566  001402                  BEQ     .+6             ;BRANCH IF OK
001570  104000                  HLT                     ;PS NOT EQUAL TO 200
001572  000026                  26                      ;THE ERROR NUMBER IS 26

001574  022767  044600  176634          CMP     #044600,ANS1    ;CHECK FIRST HALF OF ANSWER
001602  001402                  BEQ     .+6             ;BRANCH IF OK
001604  104002                  HLT+2                   ;ANS1 NOT EQUAL TO 044600
001606  000027                  27                      ;THE ERROR NUMBER IS 27

001610  005767  176624          TST     ANS2            ;CHECK SECOND HALF OF ANSWER
001614  001402                  BEQ     .+6             ;BRANCH IF OK
001616  104002                  HLT+2                   ;ANS2 NOT EQUAL TO 000000
001620  000030                  30                      ;THE ERROR NUMBER IS 30

001622  122767  000005  176554  ENDS:   CMPB    #5,     $TESTN  ;CHECK THE TEST NUMBER
001630  001402                  BEQ     .+6             ;BRANCH IF OK
001632  104000                  HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
001634  000031                  31                      ;THE ERROR NUMBER IS 31
```

```
                        ;*************************************************************************
                        ;TEST 6:        FADD (LSI-11 FLOATING ADD INSTRUCTION)
                        ;       125200,000000 + 025177,177777 = 117200,000000
                        ;       PS = 210,       STACK POINTER = SP
                        ;*************************************************************************

001636  104400                  SCOPE
001640  004567  014154  TST6:   JSR     R5,     PUSHS   ;PUSH 4 WORDS ONTO STACK, SET PRIORITY
001644  025177  177777          .WORD   025177,177777   ;SECOND OPERAND ON TOP
001650  125200  000000          .WORD   125200,000000   ;FIRST OPERAND ON BOTTOM
001654  000307                  .WORD   307             ;PROCESSOR PRIORITY LEVEL
001656  016442  000340          .WORD   TRAPER,340      ;FIS TRAP VECTOR

001662  000240                  NOP
001664  075006                  FADD    SP              ;FLOATING ADD ON THE STACK

001666  004767  014166          JSR     PC,     POPS    ;POP THE ANSWER
001672  022706  000600          CMP     #BEGIN, SP      ;CHECK THE STACK POINTER
001676  001405                  BEQ     TSA6            ;BRANCH IF OK
001700  012706  000600          MOV     #BEGIN, SP      ;RESTORE STACK POINTER
001704  104000                  HLT                     ;STACK POINTER FOULED UP
001706  000032                  32                      ;THE ERROR NUMBER IS 32
001710  000421                  BR      END6            ;SKIP REST OF TEST

001712  122767  000210  176512  TSA6:   CMPB    #210,   $PSW    ;CHECK PS (EXCEPT T BIT)
001720  001402                  BEQ     .+6             ;BRANCH IF OK
001722  104000                  HLT                     ;PS NOT EQUAL TO 210
001724  000033                  33                      ;THE ERROR NUMBER IS 33

001726  022767  117200  176502  CMP     #117200,ANS1    ;CHECK FIRST HALF OF ANSWER
001734  001402                  BEQ     .+6             ;BRANCH IF OK
001736  104002                  HLT+2                   ;ANS1 NOT EQUAL TO 117200
001740  000034                  34                      ;THE ERROR NUMBER IS 34

001742  005767  176472          TST     ANS2            ;CHECK SECOND HALF OF ANSWER
001746  001402                  BEQ     .+6             ;BRANCH IF OK
001750  104002                  HLT+2                   ;ANS2 NOT EQUAL TO 000000
001752  000035                  35                      ;THE ERROR NUMBER IS 35

001754  122767  000006  176422  END6:   CMPB    #6,     $TESTN  ;CHECK THE TEST NUMBER
001762  001402                  BEQ     .+6             ;BRANCH IF OK
001764  104000                  HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
001766  000036                  36                      ;THE ERROR NUMBER IS 36
```

```
                        ;*************************************************************************
                        ;TEST 7:        FADD (LSI-11 FLOATING ADD INSTRUCTION)
                        ;       135753,024642 + 100125,052525 = 135753,024642
                        ;       PS = 210,       STACK POINTER = R5
                        ;*************************************************************************

001770  104400                    SCOPE
001772  004767  014174    TST7:   JSR     R5,     PUSHR   ;PUSH 4 WORDS ONTO R5 STACK, SET PRIORITY
001776  100125  052525            .WORD   100125,052525   ;SECOND OPERAND ON TOP
002002  135753  024642            .WORD   135753,024642   ;FIRST OPERAND ON BOTTOM
002006  000347                    .WORD   347             ;PROCESSOR PRIORITY LEVEL
002010  016442  000340            .WORD   TRAPER,340      ;FIS TRAP VECTOR
002014  012705  000510            MOV     #STACK0,R5      ;SET UP STACK POINTER

002020  000240                    NOP
002022  075005                    FADD    R5              ;FLOATING ADD ON THE R5 STACK

002024  004767  014174            JSR     PC,     POPR    ;POP THE ANSWER
002030  010567  176400            MOV     R5,     $SP     ;SAVE "STACK POINTER"
002034  122767  000210  176370    CMPB    #210,   $PSW    ;CHECK PS (EXCEPT T BIT)
002042  001402                    BEQ     .+6             ;BRANCH IF OK
002044  104000                    HLT                     ;PS NOT EQUAL TO 210
002046  000037                    37                      ;THE ERROR NUMBER IS 37

002050  022767  000514  176356    CMP     #STACK4,$SP     ;CHECK THE STACK POINTER (R5)
002056  001402                    BEQ     .+6             ;BRANCH IF OK
002060  104000                    HLT                     ;STACK POINTER (R5) NOT EQUAL TO #STACK4
002062  000040                    40                      ;THE ERROR NUMBER IS 40

002064  022767  135753  176344    CMP     #135753,ANS1    ;CHECK FIRST HALF OF ANSWER
002072  001402                    BEQ     .+6             ;BRANCH IF OK
002074  104002                    HLT+2                   ;ANS1 NOT EQUAL TO 135753
002076  000041                    41                      ;THE ERROR NUMBER IS 41

002100  022767  024642  176332    CMP     #024642,ANS2    ;CHECK SECOND HALF OF ANSWER
002106  001402                    BEQ     .+6             ;BRANCH IF OK
002110  104002                    HLT+2                   ;ANS2 NOT EQUAL TO 024642
002112  000042                    42                      ;THE ERROR NUMBER IS 42

002114  122767  000007  176262  END7:   CMPB    #7,     $TESTN  ;CHECK THE TEST NUMBER
002122  001402                    BEQ     .+6             ;BRANCH IF OK
002124  104000                    HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
002126  000043                    43                      ;THE ERROR NUMBER IS 43
```

```
                                ;*****************************************************************************
                                ;TEST 10:       FADD (LSI-11 FLOATING ADD INSTRUCTION)
                                ;       000052,125252 + 001357,024642 = 001357,024642
                                ;       PS = 200,       STACK POINTER = R1
                                ;*****************************************************************************

002130  104400                          SCOPE
002132  004567  014034          TST10:  JSR     R5,     PUSHR   ;PUSH 4 WORDS ONTO R1 STACK, SET PRIORITY
002136  001357  024642                  .WORD   001357,024642   ;SECOND OPERAND ON TOP
002142  000052  125252                  .WORD   000052,125252   ;FIRST OPERAND ON BOTTOM
002146  000257                          .WORD   257             ;PROCESSOR PRIORITY LEVEL
002150  016442  000340                  .WORD   TRAPER,340      ;FIS TRAP VECTOR
002154  012701  000510                  MOV     #STACK0,R1      ;SET UP STACK POINTER

002160  000240                          NOP
002162  075001                          FADD    R1              ;FLOATING ADD ON THE R1 STACK

002164  004767  014034                  JSR     PC,     POPR    ;POP THE ANSWER
002170  010167  176240                  MOV     R1,     $SP     ;SAVE "STACK POINTER"
002174  122767  000200  176230          CMPB    #200,   $PSW    ;CHECK PS (EXCEPT T BIT)
002202  001402                          BEQ     .+6             ;BRANCH IF OK
002204  104000                          HLT                     ;PS NOT EQUAL TO 200
002206  000044                          44                      ;THE ERROR NUMBER IS 44

002210  022767  000514  176216          CMP     #STACK4,$SP     ;CHECK THE STACK POINTER (R1)
002216  001402                          BEQ     .+6             ;BRANCH IF OK
002220  104000                          HLT                     ;STACK POINTER (R1) NOT EQUAL TO #STACK4
002222  000045                          45                      ;THE ERROR NUMBER IS 45

002224  022767  001357  176204          CMP     #001357,ANS1    ;CHECK FIRST HALF OF ANSWER
002232  001402                          BEQ     .+6             ;BRANCH IF OK
002234  104002                          HLT+2                   ;ANS1 NOT EQUAL TO 001357
002236  000046                          46                      ;THE ERROR NUMBER IS 46

002240  022767  024642  176172          CMP     #024642,ANS2    ;CHECK SECOND HALF OF ANSWER
002246  001402                          BEQ     .+6             ;BRANCH IF OK
002250  104002                          HLT+2                   ;ANS2 NOT EQUAL TO 024642
002252  000047                          47                      ;THE ERROR NUMBER IS 47

002254  122767  000010  176122  END10:  CMPB    #10,    $TESTN  ;CHECK THE TEST NUMBER
002262  001402                          BEQ     .+6             ;BRANCH IF OK
002264  104000                          HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
002266  000050                          50                      ;THE ERROR NUMBER IS 50
```

```
                                ;*************************************************************************
                                ;TEST 11:       FADD (LSI-11 FLOATING ADD INSTRUCTION)
                                ;       100400,000000 + 000200,000000 = 100200,000000
                                ;       PS = 010,       STACK POINTER = R5
                                ;*************************************************************************

002270  104400                  SCOPE
002272  004567  013674  TST11:  JSR     R5,     PUSHR   ;PUSH 4 WORDS ONTO R5 STACK, SET PRIORITY
002276  000200  000000          .WORD   000200,000000   ;SECOND OPERAND ON TOP
002302  100400  000000          .WORD   100400,000000   ;FIRST OPERAND ON BOTTOM
002306  000140                  .WORD   140             ;PROCESSOR PRIORITY LEVEL
002310  016442  000340          .WORD   TRAPER,340      ;FIS TRAP VECTOR
002314  012705  000510          MOV     #STACK0,R5      ;SET UP STACK POINTER

002320  000240                  NOP
002322  075005                  FADD    R5              ;FLOATING ADD ON THE R5 STACK

002324  004767  013674          JSR     PC,     POPR    ;POP THE ANSWER
002330  010567  176100          MOV     R5,     $SP     ;SAVE "STACK POINTER"
002334  122767  000010  176070  CMPB    #010,   $PSW    ;CHECK PS (EXCEPT T BIT)
002342  001402                  BEQ     .+6             ;BRANCH IF OK
002344  104000                  HLT                     ;PS NOT EQUAL TO 010
002346  000051                  51                      ;THE ERROR NUMBER IS 51

002350  022767  000514  176056  CMP     #STACK4,$SP     ;CHECK THE STACK POINTER (R5)
002356  001402                  BEQ     .+6             ;BRANCH IF OK
002360  104000                  HLT                     ;STACK POINTER (R5) NOT EQUAL TO #STACK4
002362  000052                  52                      ;THE ERROR NUMBER IS 52

002364  022767  100200  176044  CMP     #100200,ANS1    ;CHECK FIRST HALF OF ANSWER
002372  001402                  BEQ     .+6             ;BRANCH IF OK
002374  104002                  HLT+2                   ;ANS1 NOT EQUAL TO 100200
002376  000053                  53                      ;THE ERROR NUMBER IS 53

002400  005767  176034          TST     ANS2            ;CHECK SECOND HALF OF ANSWER
002404  001402                  BEQ     .+6             ;BRANCH IF OK
002406  104002                  HLT+2                   ;ANS2 NOT EQUAL TO 000000
002410  000054                  54                      ;THE ERROR NUMBER IS 54

002412  122767  000011  175764  END11:  CMPB    #11,    $TESTN  ;CHECK THE TEST NUMBER
002420  001402                  BEQ     .+6             ;BRANCH IF OK
002422  104000                  HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
002424  000055                  55                      ;THE ERROR NUMBER IS 55
```

```
                        ;#########################################################################
                        ;TEST 12:       FADD (LSI-11 FLOATING ADD INSTRUCTION)
                        ;       000425,052525 + 100252,125252 = 000200,000000
                        ;       PS = 200,       STACK POINTER = R4
                        ;#########################################################################

002426  104400                    SCOPE
002430  004567  013536    TST12:  JSR     R5,     PUSHR   ;PUSH 4 WORDS ONTO R4 STACK, SET PRIORITY
002434  100252  125252            .WORD   100252,125252   ;SECOND OPERAND ON TOP
002440  000425  052525            .WORD   000425,052525   ;FIRST OPERAND ON BOTTOM
002444  000217                    .WORD   217             ;PROCESSOR PRIORITY LEVEL
002446  016442  000340            .WORD   TRAPER,340      ;FIS TRAP VECTOR
002452  012704  000510            MOV     #STACK0,R4      ;SET UP STACK POINTER

002456  000240                    NOP
002460  075004                    FADD    R4              ;FLOATING ADD ON THE R4 STACK

002462  004767  013536            JSR     PC,     POPR    ;POP THE ANSWER
002466  010467  175742            MOV     R4,     $SP     ;SAVE "STACK POINTER"
002472  122767  000200  175732    CMPB    #200,   $PSW    ;CHECK PS (EXCEPT T BIT)
002500  001402                    BEQ     .+6             ;BRANCH IF OK
002502  104000                    HLT                     ;PS NOT EQUAL TO 200
002504  000056                    56                      ;THE ERROR NUMBER IS 56

002506  022767  000514  175720    CMP     #STACK4,$SP     ;CHECK THE STACK POINTER (R4)
002514  001402                    BEQ     .+6             ;BRANCH IF OK
002516  104000                    HLT                     ;STACK POINTER (R4) NOT EQUAL TO #STACK4
002520  000057                    57                      ;THE ERROR NUMBER IS 57

002522  022767  000200  175706    CMP     #000200,ANS1    ;CHECK FIRST HALF OF ANSWER
002530  001402                    BEQ     .+6             ;BRANCH IF OK
002532  104002                    HLT+2                   ;ANS1 NOT EQUAL TO 000200
002534  000060                    60                      ;THE ERROR NUMBER IS 60

002536  005767  175676            TST     ANS2            ;CHECK SECOND HALF OF ANSWER
002542  001402                    BEQ     .+6             ;BRANCH IF OK
002544  104002                    HLT+2                   ;ANS2 NOT EQUAL TO 000000
002546  000061                    61                      ;THE ERROR NUMBER IS 61

002550  122767  000012  175626  END12:  CMPB    #12,    $TESTN  ;CHECK THE TEST NUMBER
002556  001402                    BEQ     .+6             ;BRANCH IF OK
002560  104000                    HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
002562  000062                    62                      ;THE ERROR NUMBER IS 62
```

```
                                ;*************************************************************************
                                ;TEST 13:       FADD (LSI-11 FLOATING ADD INSTRUCTION)
                                ;       100425,052525 + 000252,125252 = 100200,000000
                                ;       PS = 210,       STACK POINTER = SP
                                ;*************************************************************************

002564  104400                          SCOPE
002566  004567  013226                  TST13:  JSR     R5,     PUSHS   ;PUSH 4 WORDS ONTO STACK, SET PRIORITY
002572  000252  125252                  .WORD   000252,125252   ;SECOND OPERAND ON TOP
002576  100425  052525                  .WORD   100425,052525   ;FIRST OPERAND ON BOTTOM
002602  000307                          .WORD   307             ;PROCESSOR PRIORITY LEVEL
002604  016442  000340                  .WORD   TRAPER,340      ;FIS TRAP VECTOR

002610  000240                          NOP
002612  075006                          FADD    SP              ;FLOATING ADD ON THE STACK

002614  004767  013240                  JSR     PC,     POPS    ;POP THE ANSWER
002620  022706  000600                  CMP     #BEGIN, SP      ;CHECK THE STACK POINTER
002624  001405                          BEQ     TSA13           ;BRANCH IF OK
002626  012706  000600                  MOV     #BEGIN, SP      ;RESTORE STACK POINTER
002632  104000                          HLT                     ;STACK POINTER FOULED UP
002634  000063                          63                      ;THE ERROR NUMBER IS 63
002636  000421                          BR      END13           ;SKIP REST OF TEST

002640  122767  000210  175564  TSA13:  CMPB    #210,   $PSW    ;CHECK PS (EXCEPT T BIT)
002646  001402                          BEQ     .+6             ;BRANCH IF OK
002650  104000                          HLT                     ;PS NOT EQUAL TO 210
002652  000064                          64                      ;THE ERROR NUMBER IS 64

002654  022767  100200  175554          CMP     #100200,ANS1    ;CHECK FIRST HALF OF ANSWER
002662  001402                          BEQ     .+6             ;BRANCH IF OK
002664  104002                          HLT+2                   ;ANS1 NOT EQUAL TO 100200
002666  000065                          65                      ;THE ERROR NUMBER IS 65

002670  005767  175544                  TST     ANS2            ;CHECK SECOND HALF OF ANSWER
002674  001402                          BEQ     .+6             ;BRANCH IF OK
002676  104002                          HLT+2                   ;ANS2 NOT EQUAL TO 000000
002700  000066                          66                      ;THE ERROR NUMBER IS 66

002702  122767  000013  175474  END13:  CMPB    #13,    $TESTN  ;CHECK THE TEST NUMBER
002710  001402                          BEQ     .+6             ;BRANCH IF OK
002712  104000                          HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
002714  000067                          67                      ;THE ERROR NUMBER IS 67
```

```
                                ;*************************************************************************
                                ;TEST 14:       FADD (LSI-11 FLOATING ADD INSTRUCTION)
                                ;       077652,125252 + 077452,125252 = 077777,177777
                                ;       PS = 200,       STACK POINTER = SP
                                ;*************************************************************************

                                        SCOPE
002716  104400                  TST14:  JSR     R5,     PUSHS   ;PUSH 4 WORDS ONTO STACK, SET PRIORITY
002720  004567  013074                  .WORD   077452,125252   ;SECOND OPERAND ON TOP
002724  077452  125252                  .WORD   077652,125252   ;FIRST OPERAND ON BOTTOM
002730  077652  125252                  .WORD   257             ;PROCESSOR PRIORITY LEVEL
002734  000257                          .WORD   TRAPER,340      ;FIS TRAP VECTOR
002736  016442  000340
                                        NOP
002742  000240                          FADD    SP              ;FLOATING ADD ON THE STACK
002744  075006
                                        JSR     PC,     POPS    ;POP THE ANSWER
002746  004767  013106                  CMP     #BEGIN, SP      ;CHECK THE STACK POINTER
002752  022706  000600                  BEQ     TSA14           ;BRANCH IF OK
002756  001405                          MOV     #BEGIN, SP      ;RESTORE STACK POINTER
002760  012706  000600                  HLT                     ;STACK POINTER FOULED UP
002764  104000                          70                      ;THE ERROR NUMBER IS 70
002766  000070                          BR      END14           ;SKIP REST OF TEST
002770  000422
                                TSA14:  CMPB    #200,   $PSW    ;CHECK PS (EXCEPT T BIT)
002772  122767  000200  175432          BEQ     .+6             ;BRANCH IF OK
003000  001402                          HLT                     ;PS NOT EQUAL TO 200
003002  104000                          71                      ;THE ERROR NUMBER IS 71
003004  000071

003006  022767  077777  175422          CMP     #077777,ANS1    ;CHECK FIRST HALF OF ANSWER
003014  001402                          BEQ     .+6             ;BRANCH IF OK
003016  104002                          HLT+2                   ;ANS1 NOT EQUAL TO 077777
003020  000072                          72                      ;THE ERROR NUMBER IS 72

003022  022767  177777  175410          CMP     #177777,ANS2    ;CHECK SECOND HALF OF ANSWER
003030  001402                          BEQ     .+6             ;BRANCH IF OK
003032  104002                          HLT+2                   ;ANS2 NOT EQUAL TO 177777
003034  000073                          73                      ;THE ERROR NUMBER IS 73

003036  122767  000014  175340  END14:  CMPB    #14,    $TESTN  ;CHECK THE TEST NUMBER
003044  001402                          BEQ     .+6             ;BRANCH IF OK
003046  104000                          HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
003050  000074                          74                      ;THE ERROR NUMBER IS 74
```

```
                                ;*************************************************************************
                                ;TEST 15:       FADD (LSI-11 FLOATING ADD INSTRUCTION)
                                ;       177452,125252 + 177652,125252 = 177777,177777
                                ;       PS = 210,       STACK POINTER = R4
                                ;*************************************************************************

003052  104400                          SCOPE
003054  004567  013112          TST15:  JSR     R5,     PUSHR   ;PUSH 4 WORDS ONTO R4 STACK, SET PRIORITY
003060  177652  125252                  .WORD   177652,125252   ;SECOND OPERAND ON TOP
003064  177452  125252                  .WORD   177452,125252   ;FIRST OPERAND ON BOTTOM
003070  000357                          .WORD   357             ;PROCESSOR PRIORITY LEVEL
003072  016442  000340                  .WORD   TRAPER,340      ;FIS TRAP VECTOR
003076  012704  000510                  MOV     #STACK0,R4      ;SET UP STACK POINTER

003102  000240                          NOP
003104  075004                          FADD    R4              ;FLOATING ADD ON THE R4 STACK

003106  004767  013112                  JSR     PC,     POPR    ;POP THE ANSWER
003112  010467  175316                  MOV     R4,     $SP     ;SAVE "STACK POINTER"
003116  122767  000210  175306          CMPB    #210,   $PSW    ;CHECK PS (EXCEPT T BIT)
003124  001402                          BEQ     .+6             ;BRANCH IF OK
003126  104000                          HLT                     ;PS NOT EQUAL TO 210
003130  000075                          75                      ;THE ERROR NUMBER IS 75

003132  022767  000514  175274          CMP     #STACK4,$SP     ;CHECK THE STACK POINTER (R4)
003140  001402                          BEQ     .+6             ;BRANCH IF OK
003142  104000                          HLT                     ;STACK POINTER (R4) NOT EQUAL TO #STACK4
003144  000076                          76                      ;THE ERROR NUMBER IS 76

003146  022767  177777  175262          CMP     #177777,ANS1    ;CHECK FIRST HALF OF ANSWER
003154  001402                          BEQ     .+6             ;BRANCH IF OK
003156  104002                          HLT+2                   ;ANS1 NOT EQUAL TO 177777
003160  000077                          77                      ;THE ERROR NUMBER IS 77

003162  022767  177777  175250          CMP     #177777,ANS2    ;CHECK SECOND HALF OF ANSWER
003170  001402                          BEQ     .+6             ;BRANCH IF OK
003172  104002                          HLT+2                   ;ANS2 NOT EQUAL TO 177777
003174  000100                          100                     ;THE ERROR NUMBER IS 100

003176  122767  000015  175200  END15:  CMPB    #15,    $TESTN  ;CHECK THE TEST NUMBER
003204  001402                          BEQ     .+6             ;BRANCH IF OK
003206  104000                          HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
003210  000101                          101                     ;THE ERROR NUMBER IS 101
```

```
                                ;*************************************************************************
                                ;TEST 16:       FADD (LSI-11 FLOATING ADD INSTRUCTION)
                                ;       004000,105004 + 104000,104000 = 000401,000000
                                ;       PS = 000,       STACK POINTER = PC
                                ;*************************************************************************

                                        SCOPE
003212  104400                 TST16:  JSR     R5,     PUSH7   ;PUSH 4 WORDS ONTO STACK, SET PRIORITY
003214  004567  013124                  .WORD   STK16           ;TOP OF STACK
003220  003244                          .WORD   104000,104000   ;SECOND OPERAND ON TOP
003222  104000  104000                  .WORD   004000,105004   ;FIRST OPERAND ON BOTTOM
003226  004000  105004                  .WORD   144             ;PROCESSOR PRIORITY LEVEL
003232  000144                          .WORD   TRAPER,340      ;FIS TRAP VECTOR
003234  016442  000340

003240  000240                          NOP
003242  075007                          FADD    PC              ;FLOATING ADD ON FOLLOWING 4 WORDS
003244  104000                  STK16:  104000                  ;SHOULD CONTAIN 104000
003246  104000                          104000                  ;SHOULD CONTAIN 104000
003250  004000                          004000                  ;BEFORE FADD, 004000; AFTER, 000401
003252  105004                          105004                  ;BEFORE FADD, 105004; AFTER, 000000

003254  004767  013114                  JSR     PC,     POP7    ;POP THE ANSWER
003260  105767  175146                  TSTB    $PSW            ;CHECK PS (EXCEPT T BIT)
003264  001402                          BEQ     .+6             ;BRANCH IF OK
003266  104000                          HLT                     ;PS NOT EQUAL TO 000
003270  000102                          102                     ;THE ERROR NUMBER IS 102

003272  022767  104000  175136          CMP     #104000,ANS1    ;CHECK FIRST HALF OF INPUT DATA (STK16)
003300  001402                          BEQ     .+6             ;BRANCH IF OK
003302  104002                          HLT+2                   ;ANS1 NOT EQUAL TO 104000
003304  000103                          103                     ;THE ERROR NUMBER IS 103

003306  022767  104000  175124          CMP     #104000,ANS2    ;CHECK SECOND HALF OF INPUT DATA (STK16+2)
003314  001402                          BEQ     .+6             ;BRANCH IF OK
003316  104002                          HLT+2                   ;ANS2 NOT EQUAL TO 104000
003320  000104                          104                     ;THE ERROR NUMBER IS 104

003322  022767  000401  175112          CMP     #000401,ANS3    ;CHECK FIRST HALF OF ANSWER
003330  001402                          BEQ     .+6             ;BRANCH IF OK
003332  104004                          HLT+4                   ;ANS3 NOT EQUAL TO 000401
003334  000105                          105                     ;THE ERROR NUMBER IS 105

003336  005767  175102                  TST     ANS4            ;CHECK SECOND HALF OF ANSWER
003342  001402                          BEQ     .+6             ;BRANCH IF OK
003344  104004                          HLT+4                   ;ANS4 NOT EQUAL TO 000000
003346  000106                          106                     ;THE ERROR NUMBER IS 106

003350  122767  000016  175026  END16:  CMPB    #16,    $TESTN  ;CHECK THE TEST NUMBER
003356  001402                          BEQ     .+6             ;BRANCH IF OK
003360  104000                          HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
003362  000107                          107                     ;THE ERROR NUMBER IS 107


                                ;*************************************************************************
```

```
                                ;TEST 17:       FADD (LSI-11 FLOATING ADD INSTRUCTION)
                                ;       100200,000000 + 000377,177777 ==> UNDERFLOW
                                ;       PS(ON STACK) = 012,     STACK POINTER = R3
                                ;*********************************************************************************

003364  104400                          SCOPE
003366  004567  012600          TST17:  JSR     R5,     PUSHR   ;PUSH 4 WORDS ONTO R3 STACK, SET PRIORITY
003372  000377  177777                  .WORD   000377,177777   ;SECOND OPERAND ON TOP
003376  100200  000000                  .WORD   100200,000000   ;FIRST OPERAND ON BOTTOM
003402  000157                          .WORD   157             ;PROCESSOR PRIORITY LEVEL
003404  003436  000000                  .WORD   ISR17,  000     ;FIS TRAP VECTOR
003410  012703  000510                  MOV     #STACK0,R3      ;SET UP R3 AS STACK POINTER

003414  000240                          NOP
003416  075003                          FADD    R3              ;FLOATING ADD ON THE R3 STACK

003420  004767  012600          RTA17:  JSR     PC,     POPR    ;POP THE "ANSWER"
003424  010367  175004                  MOV     R3,     $SP     ;SAVE STACK POINTER (R3)
003430  104002                          HLT+2                   ;FIS TRAP DIDN'T OCCURE!
003432  000110                          110                     ;THE ERROR NUMBER IS 110
003434  000462                          BR      END17

003436  004767  012612          ISR17:  JSR     PC,     POPER   ;POP ALL DATA OFF THE STACKS
003442  010367  174766                  MOV     R3,     $SP     ;SAVE STACK POINTER (R3)
003446  105767  174760                  TSTB    $PSW            ;CHECK PS AFTER FIS TRAP
003452  001402                          BEQ     .+6             ;BRANCH IF OK
003454  104000                          HLT                     ;PS AFTER FIS TRAP NOT EQUAL TO 000
003456  000111                          111                     ;THE ERROR NUMBER IS 111

003460  022767  000510  174746          CMP     #STACK0,$SP     ;CHECK THE STACK POINTER (R3)
003466  001402                          BEQ     .+6             ;BRANCH IF OK
003470  104000                          HLT                     ;STACK POINTER (R3) NOT EQUAL TO #STACK0
003472  000112                          112                     ;THE ERROR NUMBER IS 112

003474  022767  003420  174734          CMP     #RTA17, ANS1    ;CHECK FIS TRAP RETURN ADDRESS
003502  001402                          BEQ     .+6             ;BRANCH IF OK
003504  104001                          HLT+1                   ;FIS TRAP AT WRONG ADDRESS
003506  000113                          113                     ;THE ERROR NUMBER IS 113

003510  022767  000012  174722          CMP     #012,   ANS2    ;CHECK PS BEFORE FIS TRAP
003516  001402                          BEQ     .+6             ;BRANCH IF OK
003520  104002                          HLT+2                   ;PS AT FIS TRAP TIME NOT 012
003522  000114                          114                     ;THE ERROR NUMBER IS 114

003524  022767  000377  174710          CMP     #000377,ANS3    ;CHECK DATA FROM THE STACK
003532  001402                          BEQ     .+6             ;BRANCH IF OK
003534  104004                          HLT+4                   ;DATA ON STACK (000377) CHANGED
003536  000115                          115                     ;THE ERROR NUMBER IS 115

003540  022767  177777  174676          CMP     #177777,ANS4    ;CHECK DATA FROM STACK
003546  001402                          BEQ     .+6             ;BRANCH IF OK
003550  104004                          HLT+4                   ;DATA ON STACK (177777) CHANGED
003552  000116                          116                     ;THE ERROR NUMBER IS 116

003554  022767  100200  174664          CMP     #100200,ANS5    ;CHECK DATA FROM STACK
003562  001402                          BEQ     .+6             ;BRANCH IF OK
```

```
 943  003564  104006                          HLT+6                   ;DATA ON STACK (100200) CHANGED
 944  003566  000117                          117                     ;THE ERROR NUMBER IS 117
 945
 946  003570  005767  174654                  TST     ANS6            ;CHECK DATA FROM STACK
 947  003574  001402                          BEQ     .+6             ;BRANCH IF OK
 948  003576  104006                          HLT+6                   ;DATA ON STACK (000000) CHANGED
 949  003600  000120                          120                     ;THE ERROR NUMBER IS 120
 950
 951  003602  122767  000017  174574  END17:  CMPB    #17,    $TESTN  ;CHECK THE TEST NUMBER
 952  003610  001402                          BEQ     .+6             ;BRANCH IF OK
 953  003612  104000                          HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
 954  003614  000121                          121                     ;THE ERROR NUMBER IS 121
 955
 956
```

```
                        ;*************************************************************************
                        ;TEST 20:       FADD (LSI-11 FLOATING ADD INSTRUCTION)
                        ;       000200,000000 + 100377,177777 ==> UNDERFLOW
                        ;       PS(ON STACK) = 212,     STACK POINTER = SP
                        ;*************************************************************************

                                SCOPE
003616  104400
003620  004567  012174  TST20:  JSR     R5,     PUSHS   ;PUSH 4 WORDS ONTO STACK, SET PRIORITY
003624  100377  177777          .WORD   100377,177777   ;SECOND OPERAND ON TOP
003630  000200  000000          .WORD   000200,000000   ;FIRST OPERAND ON BOTTOM
003634  000257                  .WORD   257             ;PROCESSOR PRIORITY LEVEL
003636  003664  000340          .WORD   ISR20,  340     ;FIS TRAP VECTOR

003642  000240                  NOP
003644  075006                  FADD    SP              ;FLOATING ADD ON THE STACK

003646  004767  012206  RTA20:  JSR     PC,     POPS    ;POP THE "ANSWER"
003652  104002                  HLT+2                   ;FIS TRAP DIDN'T OCCURE!
003654  000122                  122                     ;THE ERROR NUMBER IS 122
003656  012706  000600          MOV     #BEGIN, SP      ;RESTORE THE STACK POINTER
003662  000463                  BR      END20

003664  004767  012222  ISR20:  JSR     PC,     POPES   ;POP ALL DATA OFF THE STACK
003670  022706  000600          CMP     #BEGIN, SP      ;CHECK THE STACK POINTER
003674  001405                  BEQ     ISA20           ;BRANCH IF OK
003676  012706  000600          MOV     #BEGIN, SP      ;RESTORE THE STACK POINTER
003702  104000                  HLT                     ;STACK POINTER FOULED UP
003704  000123                  123                     ;THE ERROR NUMBER IS 123
003706  000451                  BR      END20           ;SKIP REST OF TEST

003710  122767  000340  174514  ISA20:  CMPB    #340,   $PSW    ;CHECK PS AFTER FIS TRAP
003716  001402                  BEQ     .+6             ;BRANCH IF OK
003720  104000                  HLT                     ;PS AFTER FIS TRAP NOT EQUAL TO 340
003722  000124                  124                     ;THE ERROR NUMBER IS 124

003724  022767  003646  174504          CMP     #RTA20, ANS1    ;CHECK FIS TRAP RETURN ADDRESS
003732  001402                  BEQ     .+6             ;BRANCH IF OK
003734  104001                  HLT+1                   ;FIS TRAP AT WRONG ADDRESS
003736  000125                  125                     ;THE ERROR NUMBER IS 125

003740  022767  000212  174472          CMP     #212,   ANS2    ;CHECK PS BEFORE FIS TRAP
003746  001402                  BEQ     .+6             ;BRANCH IF OK
003750  104002                  HLT+2                   ;PS AT FIS TRAP TIME NOT 212
003752  000126                  126                     ;THE ERROR NUMBER IS 126

003754  022767  100377  174460          CMP     #100377,ANS3    ;CHECK DATA FROM THE STACK
003762  001402                  BEQ     .+6             ;BRANCH IF OK
003764  104004                  HLT+4                   ;DATA ON STACK (100377) CHANGED
003766  000127                  127                     ;THE ERROR NUMBER IS 127

003770  022767  177777  174446          CMP     #177777,ANS4    ;CHECK DATA FROM STACK
003776  001402                  BEQ     .+6             ;BRANCH IF OK
004000  104004                  HLT+4                   ;DATA ON STACK (177777) CHANGED
004002  000130                  130                     ;THE ERROR NUMBER IS 130
```

```
1013  004004  022767  000200  174434          CMP     #000200,ANS5    ;CHECK DATA FROM STACK
1014  004012  001402                          BEQ     .+6             ;BRANCH IF OK
1015  004014  104006                          HLT+6                   ;DATA ON STACK (000200) CHANGED
1016  004016  000131                          131                     ;THE ERROR NUMBER IS 131
1017
1018  004020  005767  174424                  TST     ANS6            ;CHECK DATA FROM STACK
1019  004024  001402                          BEQ     .+6             ;BRANCH IF OK
1020  004026  104006                          HLT+6                   ;DATA ON STACK (000000) CHANGED
1021  004030  000132                          132                     ;THE ERROR NUMBER IS 132
1022
1023  004032  122767  000020  174344  END20:  CMPB    #20,    $TESTN  ;CHECK THE TEST NUMBER
1024  004040  001402                          BEQ     .+6             ;BRANCH IF OK
1025  004042  104000                          HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
1026  004044  000133                          133                     ;THE ERROR NUMBER IS 133
1027
1028
```

```
1029
1030                            ;*************************************************************************
1031                            ;TEST 21:       FADD (LSI-11 FLOATING ADD INSTRUCTION)
1032                            ;       177452,125253 + 177652,125252 ==> OVERFLOW
1033                            ;       PS(ON STACK) = 002,     STACK POINTER = R1
1034                            ;*************************************************************************
1035
1036 004046  104400                    SCOPE
1037 004050  004567  012116    TST21:  JSR     R5,     PUSHR   ;PUSH 4 WORDS ONTO R1 STACK, SET PRIORITY
1038 004054  177652  125252            .WORD   177652,125252   ;SECOND OPERAND ON TOP
1039 004060  177452  125253            .WORD   177452,125253   ;FIRST OPERAND ON BOTTOM
1040 004064  000105                    .WORD   105             ;PROCESSOR PRIORITY LEVEL
1041 004066  004120  000252            .WORD   ISR21,  252     ;FIS TRAP VECTOR
1042 004072  012701  000510            MOV     #STACK0,R1      ;SET UP R1 AS STACK POINTER
1043
1044 004076  000240                    NOP
1045 004100  075001                    FADD    R1              ;FLOATING ADD ON THE R1 STACK
1046
1047 004102  004767  012116    RTA21:  JSR     PC,     POPR    ;POP THE "ANSWER"
1048 004106  010167  174322            MOV     R1,     $SP     ;SAVE STACK POINTER (R1)
1049 004112  104002                    HLT+2                   ;FIS TRAP DIDN'T OCCURE!
1050 004114  000134                    134                     ;THE ERROR NUMBER IS 134
1051 004116  000464                    BR      END21
1052
1053 004120  004767  012130    ISR21:  JSR     PC,     POPER   ;POP ALL DATA OFF THE STACKS
1054 004124  010167  174304            MOV     R1,     $SP     ;SAVE STACK POINTER (R1)
1055 004130  122767  000252  174274    CMPB    #252,   $PSW    ;CHECK PS AFTER FIS TRAP
1056 004136  001402                    BEQ     .+6             ;BRANCH IF OK
1057 004140  104000                    HLT                     ;PS AFTER FIS TRAP NOT EQUAL TO 252
1058 004142  000135                    135                     ;THE ERROR NUMBER IS 135
1059
1060 004144  022767  000510  174262    CMP     #STACK0,$SP     ;CHECK THE STACK POINTER (R1)
1061 004152  001402                    BEQ     .+6             ;BRANCH IF OK
1062 004154  104000                    HLT                     ;STACK POINTER (R1) NOT EQUAL TO #STACK0
1063 004156  000136                    136                     ;THE ERROR NUMBER IS 136
1064
1065 004160  022767  004102  174250    CMP     #RTA21, ANS1    ;CHECK FIS TRAP RETURN ADDRESS
1066 004166  001402                    BEQ     .+6             ;BRANCH IF OK
1067 004170  104001                    HLT+1                   ;FIS TRAP AT WRONG ADDRESS
1068 004172  000137                    137                     ;THE ERROR NUMBER IS 137
1069
1070 004174  022767  000002  174236    CMP     #002,   ANS2    ;CHECK PS BEFORE FIS TRAP
1071 004202  001402                    BEQ     .+6             ;BRANCH IF OK
1072 004204  104002                    HLT+2                   ;PS AT FIS TRAP TIME NOT 002
1073 004206  000140                    140                     ;THE ERROR NUMBER IS 140
1074
1075 004210  022767  177652  174224    CMP     #177652,ANS3    ;CHECK DATA FROM THE STACK
1076 004216  001402                    BEQ     .+6             ;BRANCH IF OK
1077 004220  104004                    HLT+4                   ;DATA ON STACK (177652) CHANGED
1078 004222  000141                    141                     ;THE ERROR NUMBER IS 141
1079
1080 004224  022767  125252  174212    CMP     #125252,ANS4    ;CHECK DATA FROM STACK
1081 004232  001402                    BEQ     .+6             ;BRANCH IF OK
1082 004234  104004                    HLT+4                   ;DATA ON STACK (125252) CHANGED
1083 004236  000142                    142                     ;THE ERROR NUMBER IS 142
1084
```

```
1085  004240  022767  177452  174200          CMP     #177452,ANS5    ;CHECK DATA FROM STACK
1086  004246  001402                          BEQ     .+6             ;BRANCH IF OK
1087  004250  104006                          HLT+6                   ;DATA ON STACK (177452) CHANGED
1088  004252  000143                          143                     ;THE ERROR NUMBER IS 143
1089
1090  004254  022767  125253  174166          CMP     #125253,ANS6    ;CHECK DATA FROM STACK
1091  004262  001402                          BEQ     .+6             ;BRANCH IF OK
1092  004264  104006                          HLT+6                   ;DATA ON STACK (125253) CHANGED
1093  004266  000144                          144                     ;THE ERROR NUMBER IS 144
1094
1095  004270  122767  000021  174106  END21:  CMPB    #21,    $TESTN  ;CHECK THE TEST NUMBER
1096  004276  001402                          BEQ     .+6             ;BRANCH IF OK
1097  004300  104000                          HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
1098  004302  000145                          145                     ;THE ERROR NUMBER IS 145
1099
1100
```

```
1101
1102                            ;*************************************************************************
1103                            ;TEST 22:       FADD (LSI-11 FLOATING ADD INSTRUCTION)
1104                            ;       077652,125253 + 077452,125252 ==> OVERFLOW
1105                            ;       PS(ON STACK) = 002,     STACK POINTER = SP
1106                            ;*************************************************************************
1107
1108  004304  104400                       SCOPE
1109  004306  004567  011506     TST22:    JSR     R5,     PUSHS   ;PUSH 4 WORDS ONTO STACK, SET PRIORITY
1110  004312  077452  125252               .WORD   077452,125252   ;SECOND OPERAND ON TOP
1111  004316  077652  125253               .WORD   077652,125253   ;FIRST OPERAND ON BOTTOM
1112  004322  000003                       .WORD   003             ;PROCESSOR PRIORITY LEVEL
1113  004324  004352  000344               .WORD   ISR22,  344     ;FIS TRAP VECTOR
1114
1115  004330  000240                       NOP
1116  004332  075006                       FADD    SP              ;FLOATING ADD ON THE STACK
1117
1118  004334  004767  011520     RTA22:    JSR     PC,     POPS    ;POP THE "ANSWER"
1119  004340  104002                       HLT+2                   ;FIS TRAP DIDN'T OCCURE!
1120  004342  000146                       146                     ;THE ERROR NUMBER IS 146
1121  004344  012706  000600               MOV     #BEGIN, SP      ;RESTORE THE STACK POINTER
1122  004350  000464                       BR      END22
1123
1124  004352  004767  011534     ISR22:    JSR     PC,     POPES   ;POP ALL DATA OFF THE STACK
1125  004356  022706  000600               CMP     #BEGIN, SP      ;CHECK THE STACK POINTER
1126  004362  001405                       BEQ     ISA22           ;BRANCH IF OK
1127  004364  012706  000600               MOV     #BEGIN, SP      ;RESTORE THE STACK POINTER
1128  004370  104000                       HLT                     ;STACK POINTER FOULED UP
1129  004372  000147                       147                     ;THE ERROR NUMBER IS 147
1130  004374  000452                       BR      END22           ;SKIP REST OF TEST
1131
1132  004376  122767  000344  174026 ISA22:    CMPB    #344,   $PSW    ;CHECK PS AFTER FIS TRAP
1133  004404  001402                       BEQ     .+6             ;BRANCH IF OK
1134  004406  104000                       HLT                     ;PS AFTER FIS TRAP NOT EQUAL TO 344
1135  004410  000150                       150                     ;THE ERROR NUMBER IS 150
1136
1137  004412  022767  004334  174016       CMP     #RTA22, ANS1    ;CHECK FIS TRAP RETURN ADDRESS
1138  004420  001402                       BEQ     .+6             ;BRANCH IF OK
1139  004422  104001                       HLT+1                   ;FIS TRAP AT WRONG ADDRESS
1140  004424  000151                       151                     ;THE ERROR NUMBER IS 151
1141
1142  004426  022767  000002  174004       CMP     #002,   ANS2    ;CHECK PS BEFORE FIS TRAP
1143  004434  001402                       BEQ     .+6             ;BRANCH IF OK
1144  004436  104002                       HLT+2                   ;PS AT FIS TRAP TIME NOT 002
1145  004440  000152                       152                     ;THE ERROR NUMBER IS 152
1146
1147  004442  022767  077452  173772       CMP     #077452,ANS3    ;CHECK DATA FROM THE STACK
1148  004450  001402                       BEQ     .+6             ;BRANCH IF OK
1149  004452  104004                       HLT+4                   ;DATA ON STACK (077452) CHANGED
1150  004454  000153                       153                     ;THE ERROR NUMBER IS 153
1151
1152  004456  022767  125252  173760       CMP     #125252,ANS4    ;CHECK DATA FROM STACK
1153  004464  001402                       BEQ     .+6             ;BRANCH IF OK
1154  004466  104004                       HLT+4                   ;DATA ON STACK (125252) CHANGED
1155  004470  000154                       154                     ;THE ERROR NUMBER IS 154
1156
```

```
1157  004472  022767  077652  173746          CMP     #077652,ANS5    ;CHECK DATA FROM STACK
1158  004500  001402                          BEQ     .+6             ;BRANCH IF Ok
1159  004502  104006                          HLT+6                   ;DATA ON STACK (077652) CHANGED
1160  004504  000155                          155                     ;THE ERROR NUMBER IS 155
1161
1162  004506  022767  125253  173734          CMP     #125253,ANS6    ;CHECK DATA FROM STACK
1163  004514  001402                          BEQ     .+6             ;BRANCH IF OK
1164  004516  104006                          HLT+6                   ;DATA ON STACK (125253) CHANGED
1165  004520  000156                          156                     ;THE ERROR NUMBER IS 156
1166
1167  004522  122767  000022  173654  END22:  CMPB    #22,    $TESTN  ;CHECK THE TEST NUMBER
1168  004530  001402                          BEQ     .+6             ;BRANCH IF OK
1169  004532  104000                          HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
1170  004534  000157                          157                     ;THE ERROR NUMBER IS 157
1171
1172
```

```
1173
1174                                    ;*************************************************************************
1175                                    ;TEST 23:       FSUB (LSI-11 FLOATING SUBTRACT INSTRUCTION)
1176                                    ;       177520,017552 - 135352,051107 = 177520,017552
1177                                    ;       PS = 010,       STACK POINTER = R1
1178                                    ;*************************************************************************
1179
1180  004536  104400                            SCOPE
1181  004540  004567  011426           TST23:  JSR     R5,     PUSHR   ;PUSH 4 WORDS ONTO R1 STACK, SET PRIORITY
1182  004544  135352  051107                   .WORD   135352,051107   ;SECOND OPERAND ON TOP
1183  004550  177520  017552                   .WORD   177520,017552   ;FIRST OPERAND ON BOTTOM
1184  004554  000040                            .WORD   040             ;PROCESSOR PRIORITY LEVEL
1185  004556  016442  000340                   .WORD   TRAPER, 340     ;FIS TRAP VECTOR
1186  004562  012701  000510                   MOV     #STACK0,R1      ;SET UP STACK POINTER
1187
1188  004566  000240                            NOP
1189  004570  075011                            FSUB    R1              ;FLOATING SUBTRACT ON THE R1 STACK
1190
1191  004572  004767  011426                   JSR     PC,     POPR    ;POP THE ANSWER
1192  004576  010167  173632                   MOV     R1,     $SP     ;SAVE "STACK POINTER"
1193  004602  122767  000010  173622          CMPB    #010,   $PSW    ;CHECK PS (EXCEPT T BIT)
1194  004610  001402                            BEQ     .+6             ;BRANCH IF OK
1195  004612  104000                            HLT                     ;PS NOT EQUAL TO 010
1196  004614  000160                            160                     ;THE ERROR NUMBER IS 160
1197
1198  004616  022767  000514  173610           CMP     #STACK4,$SP     ;CHECK THE STACK POINTER (R1)
1199  004624  001402                            BEQ     .+6             ;BRANCH IF OK
1200  004626  104000                            HLT                     ;STACK POINTER (R1) NOT EQUAL TO #STACK4
1201  004630  000161                            161                     ;THE ERROR NUMBER IS 161
1202
1203  004632  022767  177520  173576           CMP     #177520,ANS1    ;CHECK FIRST HALF OF ANSWER
1204  004640  001402                            BEQ     .+6             ;BRANCH IF OK
1205  004642  104002                            HLT+2                   ;ANS1 NOT EQUAL TO 177520
1206  004644  000162                            162                     ;THE ERROR NUMBER IS 162
1207
1208  004646  022767  017552  173564           CMP     #017552,ANS2    ;CHECK SECOND HALF OF ANSWER
1209  004654  001402                            BEQ     .+6             ;BRANCH IF OK
1210  004656  104002                            HLT+2                   ;ANS2 NOT EQUAL TO 017552
1211  004660  000163                            163                     ;THE ERROR NUMBER IS 163
1212
1213  004662  122767  000023  173514  END23:   CMPB    #23,    $TESTN  ;CHECK THE TEST NUMBER
1214  004670  001402                            BEQ     .+6             ;BRANCH IF OK
1215  004672  104000                            HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
1216  004674  000164                            164                     ;THE ERROR NUMBER IS 164
1217
1218
```

```
1219
1220                                    ;*************************************************************************
1221                                    ;TEST 24:       FSUB (LSI-11 FLOATING SUBTRACT INSTRUCTION)
1222                                    ;       125252,125252 - 125252,125253 = 017400,000000
1223                                    ;       PS = 000,       STACK POINTER = R0
1224                                    ;*************************************************************************
1225
1226  004676  104400                            SCOPE
1227  004700  004567  011266           TST24:   JSR     R5,     PUSHR   ;PUSH 4 WORDS ONTO R0 STACK, SET PRIORITY
1228  004704  125252  125253                    .WORD   125252,125253   ;SECOND OPERAND ON TOP
1229  004710  125252  125252                    .WORD   125252,125252   ;FIRST OPERAND ON BOTTOM
1230  004714  000047                            .WORD   047             ;PROCESSOR PRIORITY LEVEL
1231  004716  016442  000340                    .WORD   TRAPER, 340     ;FIS TRAP VECTOR
1232  004722  012700  000510                    MOV     #STACK0,R0      ;SET UP STACK POINTER
1233
1234  004726  000240                            NOP
1235  004730  075010                            FSUB    R0              ;FLOATING SUBTRACT ON THE R0 STACK
1236
1237  004732  004767  011266                    JSR     PC,     POPR    ;POP THE ANSWER
1238  004736  010067  173472                    MOV     R0,     $SP     ;SAVE "STACK POINTER"
1239  004742  105767  173464                    TSTB    $PSW            ;CHECK PS (EXCEPT T BIT)
1240  004746  001402                            BEQ     .+6             ;BRANCH IF OK
1241  004750  104000                            HLT                     ;PS NOT EQUAL TO 000
1242  004752  000165                            165                     ;THE ERROR NUMBER IS 165
1243
1244  004754  022767  000514  173452            CMP     #STACK4,$SP     ;CHECK THE STACK POINTER (R0)
1245  004762  001402                            BEQ     .+6             ;BRANCH IF OK
1246  004764  104000                            HLT                     ;STACK POINTER (R0) NOT EQUAL TO #STACK4
1247  004766  000166                            166                     ;THE ERROR NUMBER IS 166
1248
1249  004770  022767  017400  173440            CMP     #017400,ANS1    ;CHECK FIRST HALF OF ANSWER
1250  004776  001402                            BEQ     .+6             ;BRANCH IF OK
1251  005000  104002                            HLT+2                   ;ANS1 NOT EQUAL TO 017400
1252  005002  000167                            167                     ;THE ERROR NUMBER IS 167
1253
1254  005004  005767  173430                    TST     ANS2            ;CHECK SECOND HALF OF ANSWER
1255  005010  001402                            BEQ     .+6             ;BRANCH IF OK
1256  005012  104002                            HLT+2                   ;ANS2 NOT EQUAL TO 000000
1257  005014  000170                            170                     ;THE ERROR NUMBER IS 170
1258
1259  005016  122767  000024  173360  END24:    CMPB    #24,    $TESTN  ;CHECK THE TEST NUMBER
1260  005024  001402                            BEQ     .+6             ;BRANCH IF OK
1261  005026  104000                            HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
1262  005030  000171                            171                     ;THE ERROR NUMBER IS 171
1263
1264
```

```
                        ;*************************************************************************
                        ;TEST 25:       FSUB (LSI-11 FLOATING SUBTRACT INSTRUCTION)
                        ;       002460,123456 - 100177,177777 = 002460,123456
                        ;       PS = 000,       STACK POINTER = SP
                        ;*************************************************************************

005032  104400                    SCOPE
005034  004567  010760  TST25:    JSR     R5,     PUSHS   ;PUSH 4 WORDS ONTO STACK, SET PRIORITY
005040  100177  177777            .WORD   100177,177777   ;SECOND OPERAND ON TOP
005044  002460  123456            .WORD   002460,123456   ;FIRST OPERAND ON BOTTOM
005050  000015                    .WORD   015             ;PROCESSOR PRIORITY LEVEL
005052  016442  000340            .WORD   TRAPER, 340     ;FIS TRAP VECTOR

005056  000240                    NOP
005060  075016                    FSUB    SP              ;FLOATING SUBTRACT ON THE STACK

005062  004767  010772            JSR     PC,     POPS    ;POP THE ANSWER
005066  022706  000600            CMP     #BEGIN, SP      ;CHECK THE STACK POINTER
005072  001405                    BEQ     TSA25           ;BRANCH IF OK
005074  012706  000600            MOV     #BEGIN, SP      ;RESTORE STACK POINTER


005100  104000                    HLT                     ;STACK POINTER FOULED UP
005102  000172                    172                     ;THE ERROR NUMBER IS 172
005104  000422                    BR      END25           ;SKIP REST OF TEST

005106  122767  000000  173316  TSA25:  CMPB    #000,   $PSW    ;CHECK PS (EXCEPT T BIT)
005114  001402                    BEQ     .+6             ;BRANCH IF OK
005116  104000                    HLT                     ;PS NOT EQUAL TO 000
005120  000173                    173                     ;THE ERROR NUMBER IS 173

005122  022767  002460  173306    CMP     #002460,ANS1    ;CHECK FIRST HALF OF ANSWER
005130  001402                    BEQ     .+6             ;BRANCH IF OK
005132  104002                    HLT+2                   ;ANS1 NOT EQUAL TO 002460
005134  000174                    174                     ;THE ERROR NUMBER IS 174

005136  022767  123456  173274    CMP     #123456,ANS2    ;CHECK SECOND HALF OF ANSWER
005144  001402                    BEQ     .+6             ;BRANCH IF OK
005146  104002                    HLT+2                   ;ANS2 NOT EQUAL TO 123456
005150  000175                    175                     ;THE ERROR NUMBER IS 175

005152  122767  000025  173224  END25:  CMPB    #25,    $TESTN  ;CHECK THE TEST NUMBER
005160  001402                    BEQ     .+6             ;BRANCH IF OK
005162  104000                    HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
005164  000176                    176                     ;THE ERROR NUMBER IS 176
```

```
                                ;*************************************************************************
                                ;TEST 26:       FSUB (LSI-11 FLOATING SUBTRACT INSTRUCTION)
                                ;       000425,052525 - 000252,125252 = 000200,000000
                                ;       PS = 200,       STACK POINTER = R4
                                ;*************************************************************************

005166  104400                          SCOPE
005170  004567  010776          TST26:  JSR     R5,     PUSHR   ;PUSH 4 WORDS ONTO R4 STACK, SET PRIORITY
005174  000252  125252                  .WORD   000252,125252   ;SECOND OPERAND ON TOP
005200  000425  052525                  .WORD   000425,052525   ;FIRST OPERAND ON BOTTOM
005204  000217                          .WORD   217             ;PROCESSOR PRIORITY LEVEL
005206  016442  000340                  .WORD   TRAPER, 340     ;FIS TRAP VECTOR
005212  012704  000510                  MOV     #STACK0,R4      ;SET UP STACK POINTER

005216  000240                          NOP
005220  075014                          FSUB    R4              ;FLOATING SUBTRACT ON THE R4 STACK

005222  004767  010776                  JSR     PC,     POPR    ;POP THE ANSWER
005226  010467  173202                  MOV     R4,     $SP     ;SAVE "STACK POINTER"
005232  122767  000200  173172          CMPB    #200,   $PSW    ;CHECK PS (EXCEPT T BIT)
005240  001402                          BEQ     .+6             ;BRANCH IF OK
005242  104000                          HLT                     ;PS NOT EQUAL TO 200
005244  000177                          177                     ;THE ERROR NUMBER IS 177

005246  022767  000514  173160          CMP     #STACK4,$SP     ;CHECK THE STACK POINTER (R4)
005254  001402                          BEQ     .+6             ;BRANCH IF OK
005256  104000                          HLT                     ;STACK POINTER (R4) NOT EQUAL TO #STACK4
005260  000200                          200                     ;THE ERROR NUMBER IS 200

005262  022767  000200  173146          CMP     #000200,ANS1    ;CHECK FIRST HALF OF ANSWER
005270  001402                          BEQ     .+6             ;BRANCH IF OK
005272  104002                          HLT+2                   ;ANS1 NOT EQUAL TO 000200
005274  000201                          201                     ;THE ERROR NUMBER IS 201

005276  005767  173136                  TST     ANS2            ;CHECK SECOND HALF OF ANSWER
005302  001402                          BEQ     .+6             ;BRANCH IF OK
005304  104002                          HLT+2                   ;ANS2 NOT EQUAL TO 000000
005306  000202                          202                     ;THE ERROR NUMBER IS 202

005310  122767  000026  173066  END26:  CMPB    #26,    $TESTN  ;CHECK THE TEST NUMBER
005316  001402                          BEQ     .+6             ;BRANCH IF OK
005320  104000                          HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
005322  000203                          203                     ;THE ERROR NUMBER IS 203
```

```
1362
1363                            ;*****************************************************************************
1364                            ;TEST 27:       FSUB (LSI-11 FLOATING SUBTRACT INSTRUCTION)
1365                            ;       177452,125252 - 077652,125252 = 177777,177777
1366                            ;       PS = 210,       STACK POINTER = SP
1367                            ;*****************************************************************************
1368
1369  005324  104400                       SCOPE
1370  005326  004567  010466     TST27:    JSR     R5,     PUSHS   ;PUSH 4 WORDS ONTO STACK, SET PRIORITY
1371  005332  077652  125252               .WORD   077652,125252   ;SECOND OPERAND ON TOP
1372  005336  177452  125252               .WORD   177452,125252   ;FIRST OPERAND ON BOTTOM
1373  005342  000357                       .WORD   357             ;PROCESSOR PRIORITY LEVEL
1374  005344  016442  000340               .WORD   TRAPER, 340     ;FIS TRAP VECTOR
1375
1376  005350  000240                       NOP
1377  005352  075016                       FSUB    SP              ;FLOATING SUBTRACT ON THE STACK
1378
1379  005354  004767  010500               JSR     PC,     POPS    ;POP THE ANSWER
1380  005360  022706  000600               CMP     #BEGIN, SP      ;CHECK THE STACK POINTER
1381  005364  001405                       BEQ     TSA27           ;BRANCH IF OK
1382  005366  012706  000600               MOV     #BEGIN, SP      ;RESTORE STACK POINTER
1383
1384
1385
1386
1387
1388  005372  104000                       HLT                     ;STACK POINTER FOULED UP
1389  005374  000204                       204                     ;THE ERROR NUMBER IS 204
1390  005376  000422                       BR      END27           ;SKIP REST OF TEST
1391
1392  005400  122767  000210  173024 TSA27:  CMPB    #210,   $PSW    ;CHECK PS (EXCEPT T BIT)
1393  005406  001402                       BEQ     .+6             ;BRANCH IF OK
1394  005410  104000                       HLT                     ;PS NOT EQUAL TO 210
1395  005412  000205                       205                     ;THE ERROR NUMBER IS 205
1396
1397  005414  022767  177777  173014       CMP     #177777,ANS1    ;CHECK FIRST HALF OF ANSWER
1398  005422  001402                       BEQ     .+6             ;BRANCH IF OK
1399  005424  104002                       HLT+2                   ;ANS1 NOT EQUAL TO 177777
1400  005426  000206                       206                     ;THE ERROR NUMBER IS 206
1401
1402  005430  022767  177777  173002       CMP     #177777,ANS2    ;CHECK SECOND HALF OF ANSWER
1403  005436  001402                       BEQ     .+6             ;BRANCH IF OK
1404  005440  104002                       HLT+2                   ;ANS2 NOT EQUAL TO 177777
1405  005442  000207                       207                     ;THE ERROR NUMBER IS 207
1406
1407  005444  122767  000027  172732 END27:  CMPB    #27,    $TESTN  ;CHECK THE TEST NUMBER
1408  005452  001402                       BEQ     .+6             ;BRANCH IF OK
1409  005454  104000                       HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
1410  005456  000210                       210                     ;THE ERROR NUMBER IS 210
1411
1412
```

```
1413
1414                                    ;*************************************************************************
1415                                    ;TEST 30:       FSUB (LSI-11 FLOATING SUBTRACT INSTRUCTION)
1416                                    ;       035152,125252 - 043125,052525 = 143125,052524
1417                                    ;       PS = 010,       STACK POINTER = R3
1418                                    ;*************************************************************************
1419
1420  005460  104400                            SCOPE
1421  005462  004567  010504           TST30:   JSR     R5,     PUSHR   ;PUSH 4 WORDS ONTO R3 STACK, SET PRIORITY
1422  005466  043125  052525                    .WORD   043125,052525   ;SECOND OPERAND ON TOP
1423  005472  035152  125252                    .WORD   035152,125252   ;FIRST OPERAND ON BOTTOM
1424  005476  000147                            .WORD   147             ;PROCESSOR PRIORITY LEVEL
1425  005500  016442  000340                    .WORD   TRAPER, 340     ;FIS TRAP VECTOR
1426  005504  012703  000510                    MOV     #STACK0,R3      ;SET UP STACK POINTER
1427
1428  005510  000240                            NOP
1429  005512  075013                            FSUB    R3              ;FLOATING SUBTRACT ON THE R3 STACK
1430
1431  005514  004767  010504                    JSR     PC,     POPR    ;POP THE ANSWER
1432  005520  010367  172710                    MOV     R3,     $SP     ;SAVE "STACK POINTER"
1433  005524  122767  000010  172700           CMPB    #010,   $PSW    ;CHECK PS (EXCEPT T BIT)
1434  005532  001402                            BEQ     .+6             ;BRANCH IF OK
1435  005534  104000                            HLT                     ;PS NOT EQUAL TO 010
1436  005536  000211                            211                     ;THE ERROR NUMBER IS 211
1437
1438  005540  022767  000514  172666            CMP     #STACK4,$SP     ;CHECK THE STACK POINTER (R3)
1439  005546  001402                            BEQ     .+6             ;BRANCH IF OK
1440  005550  104000                            HLT                     ;STACK POINTER (R3) NOT EQUAL TO #STACK4
1441  005552  000212                            212                     ;THE ERROR NUMBER IS 212
1442
1443  005554  022767  143125  172654            CMP     #143125,ANS1    ;CHECK FIRST HALF OF ANSWER
1444  005562  001402                            BEQ     .+6             ;BRANCH IF OK
1445  005564  104002                            HLT+2                   ;ANS1 NOT EQUAL TO 143125
1446  005566  000213                            213                     ;THE ERROR NUMBER IS 213
1447
1448  005570  022767  052524  172642            CMP     #052524,ANS2    ;CHECK SECOND HALF OF ANSWER
1449  005576  001402                            BEQ     .+6             ;BRANCH IF OK
1450  005600  104002                            HLT+2                   ;ANS2 NOT EQUAL TO 052524
1451  005602  000214                            214                     ;THE ERROR NUMBER IS 214
1452
1453  005604  122767  000030  172572  END30:   CMPB    #30,    $TESTN  ;CHECK THE TEST NUMBER
1454  005612  001402                            BEQ     .+6             ;BRANCH IF OK
1455  005614  104000                            HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
1456  005616  000215                            215                     ;THE ERROR NUMBER IS 215
1457
1458
```

```
1459
1460                                    ;*************************************************************************
1461                                    ;TEST 31:       FSUB (LSI-11 FLOATING SUBTRACT INSTRUCTION)
1462                                    ;       143325,052525 - 135152,125252 = 143325,052525
1463                                    ;       PS = 210,       STACK POINTER = R0
1464                                    ;*************************************************************************
1465
1466  005620  104400                            SCOPE
1467  005622  004567  010344           TST31:   JSR     R5,     PUSHR   ;PUSH 4 WORDS ONTO R0 STACK, SET PRIORITY
1468  005626  135152  125252                    .WORD   135152,125252   ;SECOND OPERAND ON TOP
1469  005632  143325  052525                    .WORD   143325,052525   ;FIRST OPERAND ON BOTTOM
1470  005636  000243                            .WORD   243             ;PROCESSOR PRIORITY LEVEL
1471  005640  016442  000340                    .WORD   TRAPER, 340     ;FIS TRAP VECTOR
1472  005644  012700  000510                    MOV     #STACK0,R0      ;SET UP STACK POINTER
1473
1474  005650  000240                            NOP
1475  005652  075010                            FSUB    R0              ;FLOATING SUBTRACT ON THE R0 STACK
1476
1477  005654  004767  010344                    JSR     PC,     POPR    ;POP THE ANSWER
1478  005660  010067  172550                    MOV     R0,     $SP     ;SAVE "STACK POINTER"
1479  005664  122767  000210  172540           CMPB    #210,   $PSW    ;CHECK PS (EXCEPT T BIT)
1480  005672  001402                            BEQ     .+6             ;BRANCH IF OK
1481  005674  104000                            HLT                     ;PS NOT EQUAL TO 210
1482  005676  000216                            216                     ;THE ERROR NUMBER IS 216
1483
1484  005700  022767  000514  172526           CMP     #STACK4,$SP     ;CHECK THE STACK POINTER (R0)
1485  005706  001402                            BEQ     .+6             ;BRANCH IF OK
1486  005710  104000                            HLT                     ;STACK POINTER (R0) NOT EQUAL TO #STACK4
1487  005712  000217                            217                     ;THE ERROR NUMBER IS 217
1488
1489  005714  022767  143325  172514           CMP     #143325,ANS1    ;CHECK FIRST HALF OF ANSWER
1490  005722  001402                            BEQ     .+6             ;BRANCH IF OK
1491  005724  104002                            HLT+2                   ;ANS1 NOT EQUAL TO 143325
1492  005726  000220                            220                     ;THE ERROR NUMBER IS 220
1493
1494  005730  022767  052525  172502           CMP     #052525,ANS2    ;CHECK SECOND HALF OF ANSWER
1495  005736  001402                            BEQ     .+6             ;BRANCH IF OK
1496  005740  104002                            HLT+2                   ;ANS2 NOT EQUAL TO 052525
1497  005742  000221                            221                     ;THE ERROR NUMBER IS 221
1498
1499  005744  122767  000031  172432  END31:   CMPB    #31,    $TESTN  ;CHECK THE TEST NUMBER
1500  005752  001402                            BEQ     .+6             ;BRANCH IF OK
1501  005754  104000                            HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
1502  005756  000222                            222                     ;THE ERROR NUMBER IS 222
1503
1504
```

```
1505
1506                                    ;*************************************************************************
1507                                    ;TEST 32:       FSUB (LSI-11 FLOATING SUBTRACT INSTRUCTION)
1508                                    ;       135152,125252 - 143325,052525 = 043325,052525
1509                                    ;       PS = 200,       STACK POINTER = R5
1510                                    ;*************************************************************************
1511
1512  005760  104400                            SCOPE
1513  005762  004567  010204           TST32:   JSR     R5,     PUSHR   ;PUSH 4 WORDS ONTO R5 STACK, SET PRIORITY
1514  005766  143325  052525                    .WORD   143325,052525   ;SECOND OPERAND ON TOP
1515  005772  135152  125252                    .WORD   135152,125252   ;FIRST OPERAND ON BOTTOM
1516  005776  000357                            .WORD   357             ;PROCESSOR PRIORITY LEVEL
1517  006000  016442  000340                    .WORD   TRAPER, 340     ;FIS TRAP VECTOR
1518  006004  012705  000510                    MOV     #STACK0,R5      ;SET UP STACK POINTER
1519
1520  006010  000240                            NOP
1521  006012  075015                            FSUB    R5              ;FLOATING SUBTRACT ON THE R5 STACK
1522
1523  006014  004767  010204                    JSR     PC,     POPR    ;POP THE ANSWER
1524  006020  010567  172410                    MOV     R5,     $SP     ;SAVE "STACK POINTER"
1525  006024  122767  000200  172400           CMPB    #200,   $PSW    ;CHECK PS (EXCEPT T BIT)
1526  006032  001402                            BEQ     .+6             ;BRANCH IF OK
1527  006034  104000                            HLT                     ;PS NOT EQUAL TO 200
1528  006036  000223                            223                     ;THE ERROR NUMBER IS 223
1529
1530  006040  022767  000514  172366           CMP     #STACK4,$SP     ;CHECK THE STACK POINTER (R5)
1531  006046  001402                            BEQ     .+6             ;BRANCH IF OK
1532  006050  104000                            HLT                     ;STACK POINTER (R5) NOT EQUAL TO #STACK4
1533  006052  000224                            224                     ;THE ERROR NUMBER IS 224
1534
1535  006054  022767  043325  172354           CMP     #043325,ANS1    ;CHECK FIRST HALF OF ANSWER
1536  006062  001402                            BEQ     .+6             ;BRANCH IF OK
1537  006064  104002                            HLT+2                   ;ANS1 NOT EQUAL TO 043325
1538  006066  000225                            225                     ;THE ERROR NUMBER IS 225
1539
1540  006070  022767  052525  172342           CMP     #052525,ANS2    ;CHECK SECOND HALF OF ANSWER
1541  006076  001402                            BEQ     .+6             ;BRANCH IF OK
1542  006100  104002                            HLT+2                   ;ANS2 NOT EQUAL TO 052525
1543  006102  000226                            226                     ;THE ERROR NUMBER IS 226
1544
1545  006104  122767  000032  172272  END32:   CMPB    #32,    $TESTN  ;CHECK THE TEST NUMBER
1546  006112  001402                            BEQ     .+6             ;BRANCH IF OK
1547  006114  104000                            HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
1548  006116  000227                            227                     ;THE ERROR NUMBER IS 227
1549
1550
```

```
1551
1552                                    ;*************************************************************************
1553                                    ;TEST 33:       FSUB (LSI-11 FLOATING SUBTRACT INSTRUCTION)
1554                                    ;       043125,052525 - 035152,125252 = 043125,052524
1555                                    ;       PS = 000,       STACK POINTER = R2
1556                                    ;*************************************************************************
1557
1558  006120  104400                            SCOPE
1559  006122  004567  010044           TST33:   JSR     R5,     PUSHR   ;PUSH 4 WORDS ONTO R2 STACK, SET PRIORITY
1560  006126  035152  125252                    .WORD   035152,125252   ;SECOND OPERAND ON TOP
1561  006132  043125  052525                    .WORD   043125,052525   ;FIRST OPERAND ON BOTTOM
1562  006136  000040                            .WORD   040             ;PROCESSOR PRIORITY LEVEL
1563  006140  016442  000340                    .WORD   TRAPER, 340     ;FIS TRAP VECTOR
1564  006144  012702  000510                    MOV     #STACK0,R2      ;SET UP STACK POINTER
1565
1566  006150  000240                            NOP
1567  006152  075012                            FSUB    R2              ;FLOATING SUBTRACT ON THE R2 STACK
1568
1569  006154  004767  010044                    JSR     PC,     POPR    ;POP THE ANSWER
1570  006160  010267  172250                    MOV     R2,     $SP     ;SAVE "STACK POINTER"
1571  006164  105767  172242                    TSTB    $PSW            ;CHECK PS (EXCEPT T BIT)
1572  006170  001402                            BEQ     .+6             ;BRANCH IF OK
1573  006172  104000                            HLT                     ;PS NOT EQUAL TO 000
1574  006174  000230                            230                     ;THE ERROR NUMBER IS 230
1575
1576  006176  022767  000514  172230            CMP     #STACK4,$SP     ;CHECK THE STACK POINTER (R2)
1577  006204  001402                            BEQ     .+6             ;BRANCH IF OK
1578  006206  104000                            HLT                     ;STACK POINTER (R2) NOT EQUAL TO #STACK4
1579  006210  000231                            231                     ;THE ERROR NUMBER IS 231
1580
1581  006212  022767  043125  172216            CMP     #043125,ANS1    ;CHECK FIRST HALF OF ANSWER
1582  006220  001402                            BEQ     .+6             ;BRANCH IF OK
1583  006222  104002                            HLT+2                   ;ANS1 NOT EQUAL TO 043125
1584  006224  000232                            232                     ;THE ERROR NUMBER IS 232
1585
1586  006226  022767  052524  172204            CMP     #052524,ANS2    ;CHECK SECOND HALF OF ANSWER
1587  006234  001402                            BEQ     .+6             ;BRANCH IF OK
1588  006236  104002                            HLT+2                   ;ANS2 NOT EQUAL TO 052524
1589  006240  000233                            233                     ;THE ERROR NUMBER IS 233
1590
1591  006242  122767  000033  172134  END33:   CMPB    #33,    $TESTN  ;CHECK THE TEST NUMBER
1592  006250  001402                            BEQ     .+6             ;BRANCH IF OK
1593  006252  104000                            HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
1594  006254  000234                            234                     ;THE ERROR NUMBER IS 234
1595
1596
```

```
1597
1598                                    ;*************************************************************************
1599                                    ;TEST 34:       FSUB (LSI-11 FLOATING SUBTRACT INSTRUCTION)
1600                                    ;       000000,000000 - 000000,000000 = 000000,000000
1601                                    ;       PS = 204,       STACK POINTER = R0
1602                                    ;*************************************************************************
1603
1604  006256  104400                            SCOPE
1605  006260  004567  007706           TST34:   JSR     R5,     PUSHR   ;PUSH 4 WORDS ONTO R0 STACK, SET PRIORITY
1606  006264  000000  000000                    .WORD   000000,000000   ;SECOND OPERAND ON TOP
1607  006270  000000  000000                    .WORD   000000,000000   ;FIRST OPERAND ON BOTTOM
1608  006274  000217                            .WORD   217             ;PROCESSOR PRIORITY LEVEL
1609  006276  016442  000340                    .WORD   TRAPER, 340     ;FIS TRAP VECTOR
1610  006302  012700  000510                    MOV     #STACK0,R0      ;SET UP STACK POINTER
1611
1612  006306  000240                            NOP
1613  006310  075010                            FSUB    R0              ;FLOATING SUBTRACT ON THE R0 STACK
1614
1615  006312  004767  007706                    JSR     PC,     POPR    ;POP THE ANSWER
1616  006316  010067  172112                    MOV     R0,     $SP     ;SAVE "STACK POINTER"
1617  006322  122767  000204  172102           CMPB    #204,   $PSW    ;CHECK PS (EXCEPT T BIT)
1618  006330  001402                            BEQ     .+6             ;BRANCH IF OK
1619  006332  104000                            HLT                     ;PS NOT EQUAL TO 204
1620  006334  000235                            235                     ;THE ERROR NUMBER IS 235
1621
1622  006336  022767  000514  172070            CMP     #STACK4,$SP     ;CHECK THE STACK POINTER (R0)
1623  006344  001402                            BEQ     .+6             ;BRANCH IF OK
1624  006346  104000                            HLT                     ;STACK POINTER (R0) NOT EQUAL TO #STACK4
1625  006350  000236                            236                     ;THE ERROR NUMBER IS 236
1626
1627  006352  005767  172060                    TST     ANS1            ;CHECK FIRST HALF OF ANSWER
1628  006356  001402                            BEQ     .+6             ;BRANCH IF OK
1629  006360  104002                            HLT+2                   ;ANS1 NOT EQUAL TO 000000
1630  006362  000237                            237                     ;THE ERROR NUMBER IS 237
1631
1632  006364  005767  172050                    TST     ANS2            ;CHECK SECOND HALF OF ANSWER
1633  006370  001402                            BEQ     .+6             ;BRANCH IF OK
1634  006372  104002                            HLT+2                   ;ANS2 NOT EQUAL TO 000000
1635  006374  000240                            240                     ;THE ERROR NUMBER IS 240
1636
1637  006376  122767  000034  172000  END34:   CMPB    #34,    $TESTN  ;CHECK THE TEST NUMBER
1638  006404  001402                            BEQ     .+6             ;BRANCH IF OK
1639  006406  104000                            HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
1640  006410  000241                            241                     ;THE ERROR NUMBER IS 241
1641
1642
```

```
                                ;*************************************************************************
                                ;TEST 35:       FSUB (LSI-11 FLOATING SUBTRACT INSTRUCTION)
                                ;       000000,000000 - 177777,177777 = 077777,177777
                                ;       PS = 000,       STACK POINTER = R2
                                ;*************************************************************************

006412  104400                          SCOPE
006414  004567  007552          TST35:  JSR     R5,     PUSHR   ;PUSH 4 WORDS ONTO R2 STACK, SET PRIORITY
006420  177777  177777                  .WORD   177777,177777   ;SECOND OPERAND ON TOP
006424  000000  000000                  .WORD   000000,000000   ;FIRST OPERAND ON BOTTOM
006430  000100                          .WORD   100             ;PROCESSOR PRIORITY LEVEL
006432  016442  000340                  .WORD   TRAPER, 340     ;FIS TRAP VECTOR
006436  012702  000510                  MOV     #STACK0,R2      ;SET UP STACK POINTER

006442  000240                          NOP
006444  075012                          FSUB    R2              ;FLOATING SUBTRACT ON THE R2 STACK

006446  004767  007552                  JSR     PC,     POPR    ;POP THE ANSWER
006452  010267  171756                  MOV     R2,     $SP     ;SAVE "STACK POINTER"
006456  105767  171750                  TSTB    $PSW            ;CHECK PS (EXCEPT T BIT)
006462  001402                          BEQ     .+6             ;BRANCH IF OK
006464  104000                          HLT                     ;PS NOT EQUAL TO 000
006466  000242                          242                     ;THE ERROR NUMBER IS 242

006470  022767  000514  171736          CMP     #STACK4,$SP     ;CHECK THE STACK POINTER (R2)
006476  001402                          BEQ     .+6             ;BRANCH IF OK
006500  104000                          HLT                     ;STACK POINTER (R2) NOT EQUAL TO #STACK4
006502  000243                          243                     ;THE ERROR NUMBER IS 243

006504  022767  077777  171724          CMP     #077777,ANS1    ;CHECK FIRST HALF OF ANSWER
006512  001402                          BEQ     .+6             ;BRANCH IF OK
006514  104002                          HLT+2                   ;ANS1 NOT EQUAL TO 077777
006516  000244                          244                     ;THE ERROR NUMBER IS 244

006520  022767  177777  171712          CMP     #177777,ANS2    ;CHECK SECOND HALF OF ANSWER
006526  001402                          BEQ     .+6             ;BRANCH IF OK
006530  104002                          HLT+2                   ;ANS2 NOT EQUAL TO 177777
006532  000245                          245                     ;THE ERROR NUMBER IS 245

006534  122767  000035  171642  END35:  CMPB    #35,    $TESTN  ;CHECK THE TEST NUMBER
006542  001402                          BEQ     .+6             ;BRANCH IF OK
006544  104000                          HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
006546  000246                          246                     ;THE ERROR NUMBER IS 246
```

```
                        ;*******************************************************************************
                        ;TEST 36:       FSUB (LSI-11 FLOATING SUBTRACT INSTRUCTION)
                        ;       000000,000000 - 077777,177777 = 177777,177777
                        ;       PS = 210,       STACK POINTER = SP
                        ;*******************************************************************************

006550  104400                    SCOPE
006552  004567  007242  TST36:    JSR     R5,     PUSHS   ;PUSH 4 WORDS ONTO STACK, SET PRIORITY
006556  077777  177777            .WORD   077777,177777   ;SECOND OPERAND ON TOP
006562  000000  000000            .WORD   000000,000000   ;FIRST OPERAND ON BOTTOM
006566  000217                    .WORD   217             ;PROCESSOR PRIORITY LEVEL
006570  016442  000340            .WORD   TRAPER, 340     ;FIS TRAP VECTOR

006574  000240                    NOP
006576  075016                    FSUB    SP              ;FLOATING SUBTRACT ON THE STACK

006600  004767  007254            JSR     PC,     POPS    ;POP THE ANSWER
006604  022706  000600            CMP     #BEGIN, SP      ;CHECK THE STACK POINTER
006610  001405                    BEQ     TSA36           ;BRANCH IF OK
006612  012706  000600            MOV     #BEGIN, SP      ;RESTORE STACK POINTER


006616  104000                    HLT                     ;STACK POINTER FOULED UP
006620  000247                    247                     ;THE ERROR NUMBER IS 247
006622  000422                    BR      END36           ;SKIP REST OF TEST

006624  122767  000210  171600  TSA36:  CMPB    #210,   $PSW    ;CHECK PS (EXCEPT T BIT)
006632  001402                    BEQ     .+6             ;BRANCH IF OK
006634  104000                    HLT                     ;PS NOT EQUAL TO 210
006636  000250                    250                     ;THE ERROR NUMBER IS 250

006640  022767  177777  171570    CMP     #177777,ANS1    ;CHECK FIRST HALF OF ANSWER
006646  001402                    BEQ     .+6             ;BRANCH IF OK
006650  104002                    HLT+2                   ;ANS1 NOT EQUAL TO 177777
006652  000251                    251                     ;THE ERROR NUMBER IS 251

006654  022767  177777  171556    CMP     #177777,ANS2    ;CHECK SECOND HALF OF ANSWER
006662  001402                    BEQ     .+6             ;BRANCH IF OK
006664  104002                    HLT+2                   ;ANS2 NOT EQUAL TO 177777
006666  000252                    252                     ;THE ERROR NUMBER IS 252

006670  122767  000036  171506  END36:  CMPB    #36,    $TESTN  ;CHECK THE TEST NUMBER
006676  001402                    BEQ     .+6             ;BRANCH IF OK
006700  104000                    HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
006702  000253                    253                     ;THE ERROR NUMBER IS 253
```

```
1740
1741                            ;*************************************************************************
1742                            ;TEST 37:       FSUB (LSI-11 FLOATING SUBTRACT INSTRUCTION)
1743                            ;       000425,052525 - 000252,125253 ==> UNDERFLOW
1744                            ;       PS(ON STACK) = 212,     STACK POINTER = SP
1745                            ;*************************************************************************
1746
1747  006704  104400                       SCOPE
1748  006706  004567  007106       TST37:  JSR     R5,     PUSHS   ;PUSH 4 WORDS ONTO STACK, SET PRIORITY
1749  006712  000252  125253               .WORD   000252,125253   ;SECOND OPERAND ON TOP
1750  006716  000425  052525               .WORD   000425,052525   ;FIRST OPERAND ON BOTTOM
1751  006722  000257                       .WORD   257             ;PROCESSOR PRIORITY LEVEL
1752  006724  006752  000340               .WORD   ISR37,  340     ;FIS TRAP VECTOR
1753
1754  006730  000240                       NOP
1755  006732  075016                       FSUB    SP              ;FLOATING SUBTRACT ON THE STACK
1756
1757  006734  004767  007120       RTA37:  JSR     PC,     POPS    ;POP THE "ANSWER"
1758  006740  104002                       HLT+2                   ;FIS TRAP DIDN'T OCCURE!
1759  006742  000254                       254                     ;THE ERROR NUMBER IS 254
1760  006744  012706  000600               MOV     #BEGIN, SP      ;RESTORE THE STACK POINTER
1761  006750  000464                       BR      END37
1762
1763  006752  004767  007134       ISR37:  JSR     PC,     POPES   ;POP ALL DATA OFF THE STACK
1764  006756  022706  000600               CMP     #BEGIN, SP      ;CHECK THE STACK POINTER
1765  006762  001405                       BEQ     ISA37           ;BRANCH IF OK
1766  006764  012706  000600               MOV     #BEGIN, SP      ;RESTORE THE STACK POINTER
1767  006770  104000                       HLT                     ;STACK POINTER FOULED UP
1768  006772  000255                       255                     ;THE ERROR NUMBER IS 255
1769  006774  000452                       BR      END37           ;SKIP REST OF TEST
1770
1771  006776  122767  000340  171426 ISA37:  CMPB    #340,   SPSW    ;CHECK PS AFTER FIS TRAP
1772  007004  001402                       BEQ     .+6             ;BRANCH IF OK
1773  007006  104000                       HLT                     ;PS AFTER FIS TRAP NOT EQUAL TO 340
1774  007010  000256                       256                     ;THE ERROR NUMBER IS 256
1775
1776  007012  022767  006734  171416       CMP     #RTA37, ANS1    ;CHECK FIS TRAP RETURN ADDRESS
1777  007020  001402                       BEQ     .+6             ;BRANCH IF OK
1778  007022  104001                       HLT+1                   ;FIS TRAP AT WRONG ADDRESS
1779  007024  000257                       257                     ;THE ERROR NUMBER IS 257
1780
1781  007026  022767  000212  171404       CMP     #212,   ANS2    ;CHECK PS BEFORE FIS TRAP
1782  007034  001402                       BEQ     .+6             ;BRANCH IF OK
1783  007036  104002                       HLT+2                   ;PS AT FIS TRAP TIME NOT 212
1784  007040  000260                       260                     ;THE ERROR NUMBER IS 260
1785
1786  007042  022767  000252  171372       CMP     #000252,ANS3    ;CHECK DATA FROM THE STACK
1787  007050  001402                       BEQ     .+6             ;BRANCH IF OK
1788  007052  104004                       HLT+4                   ;DATA ON STACK (000252) CHANGED
1789  007054  000261                       261                     ;THE ERROR NUMBER IS 261
1790
1791  007056  022767  125253  171360       CMP     #125253,ANS4    ;CHECK DATA FROM STACK
1792  007064  001402                       BEQ     .+6             ;BRANCH IF OK
1793  007066  104004                       HLT+4                   ;DATA ON STACK (125253) CHANGED
1794  007070  000262                       262                     ;THE ERROR NUMBER IS 262
1795
```

```
 1796  007072  022767  000425  171346          CMP     #000425,ANS5    ;CHECK DATA FROM STACK
 1797  007100  001402                          BEQ     .+6             ;BRANCH IF OK
 1798  007102  104006                          HLT+6                   ;DATA ON STACK (000425) CHANGED
 1799  007104  000263                          263                     ;THE ERROR NUMBER IS 263
 1800
 1801  007106  022767  052525  171334          CMP     #052525,ANS6    ;CHECK DATA FROM STACK
 1802  007114  001402                          BEQ     .+6             ;BRANCH IF OK
 1803  007116  104006                          HLT+6                   ;DATA ON STACK (052525) CHANGED
 1804  007120  000264                          264                     ;THE ERROR NUMBER IS 264
 1805
 1806  007122  122767  000037  171254  END37:  CMPB    #37,    STESTN  ;CHECK THE TEST NUMBER
 1807  007130  001402                          BEQ     .+6             ;BRANCH IF OK
 1808  007132  104000                          HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
 1809  007134  000265                          265                     ;THE ERROR NUMBER IS 265
 1810
 1811
```

```
1812
1813                                    ;*************************************************************************
1814                                    ;TEST 40:       FSUB (LSI-11 FLOATING SUBTRACT INSTRUCTION)
1815                                    ;       077652,125253 - 177452,125252 ==> OVERFLOW
1816                                    ;       PS(ON STACK) = 002,     STACK POINTER = R3
1817                                    ;*************************************************************************
1818
1819  007136  104400                            SCOPE
1820  007140  004567  007026           TST40:  JSR     R5,     PUSHR   ;PUSH 4 WORDS ONTO R3 STACK, SET PRIORITY
1821  007144  177452  125252                   .WORD   177452,125252   ;SECOND OPERAND ON TOP
1822  007150  077652  125253                   .WORD   077652,125253   ;FIRST OPERAND ON BOTTOM
1823  007154  000015                           .WORD   015             ;PROCESSOR PRIORITY LEVEL
1824  007156  007210  000344                   .WORD   ISR40, 344      ;FIS TRAP VECTOR
1825  007162  012703  000510                   MOV     #STACK0,R3      ;SET UP R3 AS STACK POINTER
1826
1827  007166  000240                           NOP
1828  007170  075013                           FSUB    R3              ;FLOATING SUBTRACT ON THE R3 STACK
1829
1830  007172  004767  007026           RTA40:  JSR     PC,     POPR    ;POP THE "ANSWER"
1831  007176  010367  171232                   MOV     R3,     $SP     ;SAVE STACK POINTER (R3)
1832  007202  104002                           HLT+2                   ;FIS TRAP DIDN'T OCCURE!
1833  007204  000266                           266                     ;THE ERROR NUMBER IS 266
1834  007206  000464                           BR      END40
1835
1836  007210  004767  007040           ISR40:  JSR     PC,     POPER   ;POP ALL DATA OFF THE STACKS
1837  007214  010367  171214                   MOV     R3,     $SP     ;SAVE STACK POINTER (R3)
1838  007220  122767  000344  171204           CMPB    #344,   $PSW    ;CHECK PS AFTER FIS TRAP
1839  007226  001402                           BEQ     .+6             ;BRANCH IF OK
1840  007230  104000                           HLT                     ;PS AFTER FIS TRAP NOT EQUAL TO 344
1841  007232  000267                           267                     ;THE ERROR NUMBER IS 267
1842
1843  007234  022767  000510  171172           CMP     #STACK0,$SP     ;CHECK THE STACK POINTER (R3)
1844  007242  001402                           BEQ     .+6             ;BRANCH IF OK
1845  007244  104000                           HLT                     ;STACK POINTER (R3) NOT EQUAL TO #STACK0
1846  007246  000270                           270                     ;THE ERROR NUMBER IS 270
1847
1848  007250  022767  007172  171160           CMP     #RTA40, ANS1    ;CHECK FIS TRAP RETURN ADDRESS
1849  007256  001402                           BEQ     .+6             ;BRANCH IF OK
1850  007260  104001                           HLT+1                   ;FIS TRAP AT WRONG ADDRESS
1851  007262  000271                           271                     ;THE ERROR NUMBER IS 271
1852
1853  007264  022767  000002  171146           CMP     #002,   ANS2    ;CHECK PS BEFORE FIS TRAP
1854  007272  001402                           BEQ     .+6             ;BRANCH IF OK
1855  007274  104002                           HLT+2                   ;PS AT FIS TRAP TIME NOT 002
1856  007276  000272                           272                     ;THE ERROR NUMBER IS 272
1857
1858  007300  022767  177452  171134           CMP     #177452,ANS3    ;CHECK DATA FROM THE STACK
1859  007306  001402                           BEQ     .+6             ;BRANCH IF OK
1860  007310  104004                           HLT+4                   ;DATA ON STACK (177452) CHANGED
1861  007312  000273                           273                     ;THE ERROR NUMBER IS 273
1862
1863  007314  022767  125252  171122           CMP     #125252,ANS4    ;CHECK DATA FROM STACK
1864  007322  001402                           BEQ     .+6             ;BRANCH IF OK
1865  007324  104004                           HLT+4                   ;DATA ON STACK (125252) CHANGED
1866  007326  000274                           274                     ;THE ERROR NUMBER IS 274
1867
```

```
 1868  007330  022767  077652  171110          CMP     #077652,ANS5    ;CHECK DATA FROM STACK
 1869  007336  001402                          BEQ     .+6             ;BRANCH IF OK
 1870  007340  104006                          HLT+6                   ;DATA ON STACK (077652) CHANGED
 1871  007342  000275                          275                     ;THE ERROR NUMBER IS 275
 1872
 1873  007344  022767  125253  171076          CMP     #125253,ANS6    ;CHECK DATA FROM STACK
 1874  007352  001402                          BEQ     .+6             ;BRANCH IF OK
 1875  007354  104006                          HLT+6                   ;DATA ON STACK (125253) CHANGED
 1876  007356  000276                          276                     ;THE ERROR NUMBER IS 276
 1877
 1878  007360  122767  000040  171016  END40:  CMPB    #40,    $TESTN  ;CHECK THE TEST NUMBER
 1879  007366  001402                          BEQ     .+6             ;BRANCH IF OK
 1880  007370  104000                          HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
 1881  007372  000277                          277                     ;THE ERROR NUMBER IS 277
 1882
 1883
```

```
                                ;*************************************************************************
                                ;TEST 41:       FMUL (LSI-11 FLOATING MULTIPLY INSTRUCTION)
                                ;       000000,000000 * 000000,000000 = 000000,000000
                                ;       PS = 004,       STACK POINTER = R4
                                ;*************************************************************************

007374  104400                          SCOPE
007376  004567  006570          TST41:  JSR     R5,     PUSHR   ;PUSH 4 WORDS ONTO R4 STACK, SET PRIORITY
007402  000000  000000                  .WORD   000000,000000   ;SECOND OPERAND ON TOP
007406  000000  000000                  .WORD   000000,000000   ;FIRST OPERAND ON BOTOM
007412  000111                          .WORD   111             ;PROCESSOR PRIORITY LEVEL
007414  016442  000340                  .WORD   TRAPER, 340     ;FIS TRAP VECTOR
007420  012704  000510                  MOV     #STACK0,R4      ;SET UP STACK POINTER

007424  000240                          NOP
007426  075024                          FMUL    R4              ;FLOATING MULTIPLY ON THE R4 STACK

007430  004767  006570                  JSR     PC,     POPR    ;POP THE ANSWER
007434  010467  170774                  MOV     R4,     $SP     ;SAVE "STACK POINTER"
007440  122767  000004  170764          CMPB    #004,   $PSW    ;CHECK PS (EXCEPT T BIT)
007446  001402                          BEQ     .+6             ;BRANCH IF OK
007450  104000                          HLT                     ;PS NOT EQUAL TO 004
007452  000300                          300                     ;THE ERROR NUMBER IS 300

007454  022767  000514  170752          CMP     #STACK4,$SP     ;CHECK THE STACK POINTER (R4)
007462  001402                          BEQ     .+6             ;BRANCH IF OK
007464  104000                          HLT                     ;STACK POINTER (R4) NOT EQUAL TO #STACK4
007466  000301                          301                     ;THE ERROR NUMBER IS 301

007470  005767  170742                  TST     ANS1            ;CHECK FIRST HALF OF ANSWER
007474  001402                          BEQ     .+6             ;BRANCH IF OK
007476  104002                          HLT+2                   ;ANS1 NOT EQUAL TO 000000
007500  000302                          302                     ;THE ERROR NUMBER IS 302

007502  005767  170732                  TST     ANS2            ;CHECK SECOND HALF OF ANSWER
007506  001402                          BEQ     .+6             ;BRANCH IF OK
007510  104002                          HLT+2                   ;ANS2 NOT EQUAL TO 000000
007512  000303                          303                     ;THE ERROR NUMBER IS 303

007514  122767  000041  170662  END41:  CMPB    #41,    $TESTN  ;CHECK THE TEST NUMBER
007522  001402                          BEQ     .+6             ;BRANCH IF OK
007524  104000                          HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
007526  000304                          304                     ;THE ERROR NUMBER IS 304
```

```
                                ;*************************************************************************
                                ;TEST 42:       FMUL (LSI-11 FLOATING MULTIPLY INSTRUCTION)
                                ;       140200,000000 * 052345,123456 = 152345,123456
                                ;       PS = 210,       STACK POINTER = R2
                                ;*************************************************************************

007530  104400                          SCOPE
007532  004567  006434        TST42:  JSR     R5,     PUSHR   ;PUSH 4 WORDS ONTO R2 STACK, SET PRIORITY
007536  052345  123456                .WORD   052345,123456   ;SECOND OPERAND ON TOP
007542  140200  000000                .WORD   140200,000000   ;FIRST OPERAND ON BOTOM
007546  000343                        .WORD   343             ;PROCESSOR PRIORITY LEVEL
007550  016442  000340                .WORD   TRAPER, 340     ;FIS TRAP VECTOR
007554  012702  000510                MOV     #STACK0,R2      ;SET UP STACK POINTER

007560  000240                        NOP
007562  075022                        FMUL    R2              ;FLOATING MULTIPLY ON THE R2 STACK

007564  004767  006434                JSR     PC,     POPR    ;POP THE ANSWER
007570  010267  170640                MOV     R2,     $SP     ;SAVE "STACK POINTER"
007574  122767  000210  170630        CMPB    #210,   $PSW    ;CHECK PS (EXCEPT T BIT)
007602  001402                        BEQ     .+6             ;BRANCH IF OK
007604  104000                        HLT                     ;PS NOT EQUAL TO 210
007606  000305                        305                     ;THE ERROR NUMBER IS 305

007610  022767  000514  170616        CMP     #STACK4,$SP     ;CHECK THE STACK POINTER (R2)
007616  001402                        BEQ     .+6             ;BRANCH IF OK
007620  104000                        HLT                     ;STACK POINTER (R2) NOT EQUAL TO #STACK4
007622  000306                        306                     ;THE ERROR NUMBER IS 306

007624  022767  152345  170604        CMP     #152345,ANS1    ;CHECK FIRST HALF OF ANSWER
007632  001402                        BEQ     .+6             ;BRANCH IF OK
007634  104002                        HLT+2                   ;ANS1 NOT EQUAL TO 152345
007636  000307                        307                     ;THE ERROR NUMBER IS 307

007640  022767  123456  170572        CMP     #123456,ANS2    ;CHECK SECOND HALF OF ANSWER
007646  001402                        BEQ     .+6             ;BRANCH IF OK
007650  104002                        HLT+2                   ;ANS2 NOT EQUAL TO 123456
007652  000310                        310                     ;THE ERROR NUMBER IS 310

007654  122767  000042  170522  END42:  CMPB    #42,    $TESTN  ;CHECK THE TEST NUMBER
007662  001402                        BEQ     .+6             ;BRANCH IF OK
007664  104000                        HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
007666  000311                        311                     ;THE ERROR NUMBER IS 311
```

```
                                ;*****************************************************************************
                                ;TEST 43:       FMUL (LSI-11 FLOATING MULTIPLY INSTRUCTION)
                                ;       100125,052525 * 135753,024642 = 000000,000000
                                ;       PS = 004,       STACK POINTER = R5
                                ;*****************************************************************************

007670  104400                          SCOPE
007672  004567  006274          TST43:  JSR     R5,     PUSHR   ;PUSH 4 WORDS ONTO R5 STACK, SET PRIORITY
007676  135753  024642                  .WORD   135753,024642   ;SECOND OPERAND ON TOP
007702  100125  052525                  .WORD   100125,052525   ;FIRST OPERAND ON BOTOM
007706  000117                          .WORD   117             ;PROCESSOR PRIORITY LEVEL
007710  016442  000340                  .WORD   TRAPER, 340     ;FIS TRAP VECTOR
007714  012705  000510                  MOV     #STACK0,R5      ;SET UP STACK POINTER

007720  000240                          NOP
007722  075025                          FMUL    R5              ;FLOATING MULTIPLY ON THE R5 STACK

007724  004767  006274                  JSR     PC,     POPR    ;POP THE ANSWER
007730  010567  170500                  MOV     R5,     $SP     ;SAVE "STACK POINTER"
007734  122767  000004  170470          CMPB    #004,   $PSW    ;CHECK PS (EXCEPT T BIT)
007742  001402                          BEQ     .+6             ;BRANCH IF OK
007744  104000                          HLT                     ;PS NOT EQUAL TO 004
007746  000312                          312                     ;THE ERROR NUMBER IS 312

007750  022767  000514  170456          CMP     #STACK4,$SP     ;CHECK THE STACK POINTER (R5)
007756  001402                          BEQ     .+6             ;BRANCH IF OK
007760  104000                          HLT                     ;STACK POINTER (R5) NOT EQUAL TO #STACK4
007762  000313                          313                     ;THE ERROR NUMBER IS 313

007764  005767  170446                  TST     ANS1            ;CHECK FIRST HALF OF ANSWER
007770  001402                          BEQ     .+6             ;BRANCH IF OK
007772  104002                          HLT+2                   ;ANS1 NOT EQUAL TO 000000
007774  000314                          314                     ;THE ERROR NUMBER IS 314

007776  005767  170436                  TST     ANS2            ;CHECK SECOND HALF OF ANSWER
010002  001402                          BEQ     .+6             ;BRANCH IF OK
010004  104002                          HLT+2                   ;ANS2 NOT EQUAL TO 000000
010006  000315                          315                     ;THE ERROR NUMBER IS 315

010010  122767  000043  170366  END43:  CMPB    #43,    $TESTN  ;CHECK THE TEST NUMBER
010016  001402                          BEQ     .+6             ;BRANCH IF OK
010020  104000                          HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
010022  000316                          316                     ;THE ERROR NUMBER IS 316
```

```
2022
2023                            ;*************************************************************************
2024                            ;TEST 44:       FMUL (LSI-11 FLOATING MULTIPLY INSTRUCTION)
2025                            ;       161616,161616 * 000052,125252 = 000000,000000
2026                            ;       PS = 204,       STACK POINTER = R3
2027                            ;*************************************************************************
2028
2029  010024  104400                           SCOPE
2030  010026  004567  006140           TST44:  JSR     R5,     PUSHR   ;PUSH 4 WORDS ONTO R3 STACK, SET PRIORITY
2031  010032  000052  125252                   .WORD   000052,125252   ;SECOND OPERAND ON TOP
2032  010036  161616  161616                   .WORD   161616,161616   ;FIRST OPERAND ON BOTOM
2033  010042  000217                           .WORD   217             ;PROCESSOR PRIORITY LEVEL
2034  010044  016442  000340                   .WORD   TRAPER, 340     ;FIS TRAP VECTOR
2035  010050  012703  000510                   MOV     #STACK0,R3      ;SET UP STACK POINTER
2036
2037  010054  000240                           NOP
2038  010056  075023                           FMUL    R3              ;FLOATING MULTIPLY ON THE R3 STACK
2039
2040  010060  004767  006140                   JSR     PC,     POPR    ;POP THE ANSWER
2041  010064  010367  170344                   MOV     R3,     $SP     ;SAVE "STACK POINTER"
2042  010070  122767  000204  170334           CMPB    #204,   $PSW    ;CHECK PS (EXCEPT T BIT)
2043  010076  001402                           BEQ     .+6             ;BRANCH IF OK
2044  010100  104000                           HLT                     ;PS NOT EQUAL TO 204
2045  010102  000317                           317                     ;THE ERROR NUMBER IS 317
2046
2047  010104  022767  000514  170322           CMP     #STACK4,$SP     ;CHECK THE STACK POINTER (R3)
2048  010112  001402                           BEQ     .+6             ;BRANCH IF OK
2049  010114  104000                           HLT                     ;STACK POINTER (R3) NOT EQUAL TO #STACK4
2050  010116  000320                           320                     ;THE ERROR NUMBER IS 320
2051
2052  010120  005767  170312                   TST     ANS1            ;CHECK FIRST HALF OF ANSWER
2053  010124  001402                           BEQ     .+6             ;BRANCH IF OK
2054  010126  104002                           HLT+2                   ;ANS1 NOT EQUAL TO 000000
2055  010130  000321                           321                     ;THE ERROR NUMBER IS 321
2056
2057  010132  005767  170302                   TST     ANS2            ;CHECK SECOND HALF OF ANSWER
2058  010136  001402                           BEQ     .+6             ;BRANCH IF OK
2059  010140  104002                           HLT+2                   ;ANS2 NOT EQUAL TO 000000
2060  010142  000322                           322                     ;THE ERROR NUMBER IS 322
2061
2062  010144  122767  000044  170232  END44:  CMPB    #44,    $TESTN  ;CHECK THE TEST NUMBER
2063  010152  001402                           BEQ     .+6             ;BRANCH IF OK
2064  010154  104000                           HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
2065  010156  000323                           323                     ;THE ERROR NUMBER IS 323
2066
2067
```

```
2068
2069                            ;*****************************************************************************
2070                            ;TEST 45:       FMUL (LSI-11 FLOATING MULTIPLY INSTRUCTION)
2071                            ;       176452,125252 * 041500,000000 = 177777,177777
2072                            ;       PS = 210,       STACK POINTER = SP
2073                            ;*****************************************************************************
2074
2075  010160  104400                       SCOPE
2076  010162  004567  005632       TST45:  JSR     R5,     PUSHS   ;PUSH 4 WORDS ONTO STACK, SET PRIORITY
2077  010166  041500  000000               .WORD   041500,000000   ;SECOND OPERAND ON TOP
2078  010172  176452  125252               .WORD   176452,125252   ;FIRST OPERAND ON BOTTOM
2079  010176  000357                       .WORD   357             ;PROCESSOR PRIORITY LEVEL
2080  010200  016442  000340               .WORD   TRAPER, 340     ;FIS TRAP VECTOR
2081
2082  010204  000240                       NOP
2083  010206  075026                       FMUL    SP              ;FLOATING MULTIPLY ON THE STACK
2084
2085  010210  004767  005644               JSR     PC,     POPS    ;POP THE ANSWER
2086  010214  022706  000600               CMP     #BEGIN, SP      ;CHECK THE STACK POINTER
2087  010220  001405                       BEQ     TSA45           ;BRANCH IF OK
2088  010222  012706  000600               MOV     #BEGIN, SP      ;RESTORE STACK POINTER
2089  010226  104000                       HLT                     ;STACK POINTER FOULED UP
2090  010230  000324                       324                     ;THE ERROR NUMBER IS 324
2091  010232  000422                       BR      END45           ;SKIP REST OF TEST
2092
2093  010234  122767  000210  170170 TSA45:  CMPB    #210,   $PSW    ;CHECK PS (EXCEPT T BIT)
2094  010242  001402                       BEQ     .+6             ;BRANCH IF OK
2095  010244  104000                       HLT                     ;PS NOT EQUAL TO 210
2096  010246  000325                       325                     ;THE ERROR NUMBER IS 325
2097
2098  010250  022767  177777  170160       CMP     #177777,ANS1    ;CHECK FIRST HALF OF ANSWER
2099  010256  001402                       BEQ     .+6             ;BRANCH IF OK
2100  010260  104002                       HLT+2                   ;ANS1 NOT EQUAL TO 177777
2101  010262  000326                       326                     ;THE ERROR NUMBER IS 326
2102
2103  010264  022767  177777  170146       CMP     #177777,ANS2    ;CHECK SECOND HALF OF ANSWER
2104  010272  001402                       BEQ     .+6             ;BRANCH IF OK
2105  010274  104002                       HLT+2                   ;ANS2 NOT EQUAL TO 177777
2106  010276  000327                       327                     ;THE ERROR NUMBER IS 327
2107
2108  010300  122767  000045  170076 END45:  CMPB    #45,    $TESTN  ;CHECK THE TEST NUMBER
2109  010306  001402                       BEQ     .+6             ;BRANCH IF OK
2110  010310  104000                       HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
2111  010312  000330                       330.                    ;THE ERROR NUMBER IS 330
2112
2113
```

```
2114
2115                                    ;*************************************************************************
2116                                    ;TEST 46:       FMUL (LSI-11 FLOATING MULTIPLY INSTRUCTION)
2117                                    ;       124252,125252 * 114100,000001 = 000200,000000
2118                                    ;       PS = 200,       STACK POINTER = R1
2119                                    ;*************************************************************************
2120
2121  010314  104400                            SCOPE
2122  010316  004567  005650           TST46:  JSR     R5,     PUSHR   ;PUSH 4 WORDS ONTO R1 STACK, SET PRIORITY
2123  010322  114100  000001                   .WORD   114100,000001   ;SECOND OPERAND ON TOP
2124  010326  124252  125252                   .WORD   124252,125252   ;FIRST OPERAND ON BOTOM
2125  010332  000200                            .WORD   200             ;PROCESSOR PRIORITY LEVEL
2126  010334  016442  000340                   .WORD   TRAPER, 340     ;FIS TRAP VECTOR
2127  010340  012701  000510                   MOV     #STACK0,R1      ;SET UP STACK POINTER
2128
2129  010344  000240                            NOP
2130  010346  075021                            FMUL    R1              ;FLOATING MULTIPLY ON THE R1 STACK
2131
2132  010350  004767  005650                    JSR     PC,     POPR    ;POP THE ANSWER
2133  010354  010167  170054                    MOV     R1,     $SP     ;SAVE "STACK POINTER"
2134  010360  122767  000200  170044           CMPB    #200,   $PSW    ;CHECK PS (EXCEPT T BIT)
2135  010366  001402                            BEQ     .+6             ;BRANCH IF OK
2136  010370  104000                            HLT                     ;PS NOT EQUAL TO 200
2137  010372  000331                            331                     ;THE ERROR NUMBER IS 331
2138
2139  010374  022767  000514  170032           CMP     #STACK4,$SP     ;CHECK THE STACK POINTER (R1)
2140  010402  001402                            BEQ     .+6             ;BRANCH IF OK
2141  010404  104000                            HLT                     ;STACK POINTER (R1) NOT EQUAL TO #STACK4
2142  010406  000332                            332                     ;THE ERROR NUMBER IS 332
2143
2144  010410  022767  000200  170020            CMP     #000200,ANS1    ;CHECK FIRST HALF OF ANSWER
2145  010416  001402                            BEQ     .+6             ;BRANCH IF OK
2146  010420  104002                            HLT+2                   ;ANS1 NOT EQUAL TO 000200
2147  010422  000333                            333                     ;THE ERROR NUMBER IS 333
2148
2149  010424  005767  170010                    TST     ANS2            ;CHECK SECOND HALF OF ANSWER
2150  010430  001402                            BEQ     .+6             ;BRANCH IF OK
2151  010432  104002                            HLT+2                   ;ANS2 NOT EQUAL TO 000000
2152  010434  000334                            334                     ;THE ERROR NUMBER IS 334
2153
2154  010436  122767  000046  167740  END46:   CMPB    #46,    $TESTN  ;CHECK THE TEST NUMBER
2155  010444  001402                            BEQ     .+6             ;BRANCH IF OK
2156  010446  104000                            HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
2157  010450  000335                            335                     ;THE ERROR NUMBER IS 335
2158
2159
```

```
2160
2161                            ;*************************************************************************
2162                            ;TEST 47:       FSUB (LSI-11 FLOATING SUBTRACT INSTRUCTION)
2163                            ;       104000,105004 - 104000,104000 = 100401,000000
2164                            ;       PS = 210,       STACK POINTER = PC
2165                            ;*************************************************************************
2166
2167 010452  104400                    SCOPE
2168 010454  004567  005664    TST47:  JSR     R5,     PUSH7   ;PUSH 4 WORDS ONTO STACK, SET PRIORITY
2169 010460  010504                    .WORD   STK47           ;TOP OF STACK
2170 010462  104000  104000            .WORD   104000,104000   ;SECOND OPERAND ON TOP
2171 010466  104000  105004            .WORD   104000,105004   ;FIRST OPERAND ON BOTTOM
2172 010472  000252                    .WORD   252             ;PROCESSOR PRIORITY LEVEL
2173 010474  016442  000340            .WORD   TRAPER,340      ;FIS TRAP VECTOR
2174
2175 010500  000240                    NOP
2176 010502  075017                    FSUB    PC              ;FLOATING SUBTRACT ON FOLLOWING 4 WORDS
2177 010504  104000    STK47:  104000                          ;SHOULD CONTAIN 104000
2178 010506  104000                    104000                  ;SHOULD CONTAIN 104000
2179 010510  104000                    104000                  ;BEFORE FSUB, 104000; AFTER, 100401
2180 010512  105004                    105004                  ;BEFORE FSUB, 105004; AFTER, 000000
2181
2182 010514  004767  005654            JSR     PC,     POP7    ;POP THE ANSWER
2183 010520  122767  000210  167704    CMPB    #210,   $PSW    ;CHECK PS (EXCEPT T BIT)
2184 010526  001402                    BEQ     .+6             ;BRANCH IF OK
2185 010530  104000                    HLT                     ;PS NOT EQUAL TO 210
2186 010532  000336                    336                     ;THE ERROR NUMBER IS 336
2187
2188 010534  022767  104000  167674    CMP     #104000,ANS1    ;CHECK FIRST HALF OF INPUT DATA (STK47)
2189 010542  001402                    BEQ     .+6             ;BRANCH IF OK
2190 010544  104002                    HLT+2                   ;ANS1 NOT EQUAL TO 104000
2191 010546  000337                    337                     ;THE ERROR NUMBER IS 337
2192
2193 010550  022767  104000  167662    CMP     #104000,ANS2    ;CHECK SECOND HALF OF INPUT DATA (STK47+2)
2194 010556  001402                    BEQ     .+6             ;BRANCH IF OK
2195 010560  104002                    HLT+2                   ;ANS2 NOT EQUAL TO 104000
2196 010562  000340                    340                     ;THE ERROR NUMBER IS 340
2197
2198 010564  022767  100401  167650    CMP     #100401,ANS3    ;CHECK FIRST HALF OF ANSWER
2199 010572  001402                    BEQ     .+6             ;BRANCH IF OK
2200 010574  104004                    HLT+4                   ;ANS3 NOT EQUAL TO 100401
2201 010576  000341                    341                     ;THE ERROR NUMBER IS 341
2202
2203 010600  005767  167640            TST     ANS4            ;CHECK SECOND HALF OF ANSWER
2204 010604  001402                    BEQ     .+6             ;BRANCH IF OK
2205 010606  104004                    HLT+4                   ;ANS4 NOT EQUAL TO 000000
2206 010610  000342                    342                     ;THE ERROR NUMBER IS 342
2207
2208 010612  122767  000047  167564  END47:  CMPB    #47,    $TESTN  ;CHECK THE TEST NUMBER
2209 010620  001402                    BEQ     .+6             ;BRANCH IF OK
2210 010622  104000                    HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
2211 010624  000343                    343                     ;THE ERROR NUMBER IS 343
2212
2213
```

```
                        ;*************************************************************************
                        ;TEST 50:       FMUL (LSI-11 FLOATING MULTIPLY INSTRUCTION)
                        ;       134600,073601 * 104000,104000 = 000401,000000
                        ;       PS = 200,       STACK POINTER = PC
                        ;*************************************************************************

010626  104400                    SCOPE
010630  004567  005510    TST50:  JSR     R5,     PUSH7   ;PUSH 4 WORDS ONTO STACK, SET PRIORITY
010634  010660                    .WORD   STK50           ;TOP OF STACK
010636  104000  104000            .WORD   104000,104000   ;SECOND OPERAND ON TOP
010642  134600  073601            .WORD   134600,073601   ;FIRST OPERAND ON BOTTOM
010646  000246                    .WORD   246             ;PROCESSOR PRIORITY LEVEL
010650  016442  000340            .WORD   TRAPER,340      ;FIS TRAP VECTOR

010654  000240                    NOP
010656  075027                    FMUL    PC              ;FLOATING MULTIPLY ON FOLLOWING 4 WORDS
010660  104000    STK50:  104000          ;SHOULD CONTAIN 104000
010662  104000            104000          ;SHOULD CONTAIN 104000
010664  134600            134600          ;BEFORE FMUL, 134600; AFTER, 000401
010666  073601            073601          ;BEFORE FMUL, 073601; AFTER, 000000

010670  004767  005500            JSR     PC,     POP7    ;POP THE ANSWER
010674  122767  000200  167530    CMPB    #200,   $PSW    ;CHECK PS (EXCEPT T BIT)
010702  001402                    BEQ     .+6             ;BRANCH IF OK
010704  104000                    HLT                     ;PS NOT EQUAL TO 200
010706  000344                    344                     ;THE ERROR NUMBER IS 344

010710  022767  104000  167520    CMP     #104000,ANS1    ;CHECK FIRST HALF OF INPUT DATA (STK50)
010716  001402                    BEQ     .+6             ;BRANCH IF OK
010720  104002                    HLT+2                   ;ANS1 NOT EQUAL TO 104000
010722  000345                    345                     ;THE ERROR NUMBER IS 345

010724  022767  104000  167506    CMP     #104000,ANS2    ;CHECK SECOND HALF OF INPUT DATA (STK50+2)
010732  001402                    BEQ     .+6             ;BRANCH IF OK
010734  104002                    HLT+2                   ;ANS2 NOT EQUAL TO 104000
010736  000346                    346                     ;THE ERROR NUMBER IS 346

010740  022767  000401  167474    CMP     #000401,ANS3    ;CHECK FIRST HALF OF ANSWER
010746  001402                    BEQ     .+6             ;BRANCH IF OK
010750  104004                    HLT+4                   ;ANS3 NOT EQUAL TO 000401
010752  000347                    347                     ;THE ERROR NUMBER IS 347

010754  005767  167464            TST     ANS4            ;CHECK SECOND HALF OF ANSWER
010760  001402                    BEQ     .+6             ;BRANCH IF OK
010762  104004                    HLT+4                   ;ANS4 NOT EQUAL TO 000000
010764  000350                    350                     ;THE ERROR NUMBER IS 350

010766  122767  000050  167410  END50:  CMPB    #50,    $TESTN  ;CHECK THE TEST NUMBER
010774  001402                    BEQ     .+6             ;BRANCH IF OK
010776  104000                    HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
011000  000351                    351                     ;THE ERROR NUMBER IS 351


                        ;*************************************************************************
```

```
2270                                    ;TEST 51:       FMUL (LSI-11 FLOATING MULTIPLY INSTRUCTION)
2271                                    ;       024252,125252 * 114100,000000 ==> UNDERFLOW
2272                                    ;       PS(ON STACK) = 212,     STACK POINTER = R0
2273                                    ;*********************************************************************
2274
2275  011002  104400                           SCOPE
2276  011004  004567  005162           TST51:  JSR     R5,     PUSHR   ;PUSH 4 WORDS ONTO R0 STACK, SET PRIORITY
2277  011010  114100  000000                   .WORD   114100,000000   ;SECOND OPERAND ON TOP
2278  011014  024252  125252                   .WORD   024252,125252   ;FIRST OPERAND ON BOTTOM
2279  011020  000305                           .WORD   305             ;PROCESSOR PRIORITY LEVEL
2280  011022  011054  000057                   .WORD   ISR51, 057      ;FIS TRAP VECTOR
2281  011026  012700  000510                   MOV     #STACK0,R0      ;SET UP R0 AS STACK POINTER
2282
2283  011032  000240                           NOP
2284  011034  075020                           FMUL    R0              ;FLOATING MULTIPLY ON THE R0 STACK
2285
2286  011036  004767  005162           RTA51:  JSR     PC,     POPR    ;POP THE "ANSWER"
2287  011042  010067  167366                   MOV     R0,     $SP     ;SAVE STACK POINTER (R0)
2288  011046  104002                           HLT+2                   ;FIS TRAP DIDN'T OCCURE!
2289  011050  000352                           352                     ;THE ERROR NUMBER IS 352
2290  011052  000463                           BR      END51
2291
2292  011054  004767  005174           ISR51:  JSR     PC,     POPER   ;POP ALL DATA OFF THE STACKS
2293  011060  010067  167350                   MOV     R0,     $SP     ;SAVE STACK POINTER (R0)
2294  011064  122767  000057  167340           CMPB    #057,   $PSW    ;CHECK PS AFTER FIS TRAP
2295  011072  001402                           BEQ     .+6             ;BRANCH IF OK
2296  011074  104000                           HLT                     ;PS AFTER FIS TRAP NOT EQUAL TO 057
2297  011076  000353                           353                     ;THE ERROR NUMBER IS 353
2298
2299  011100  022767  000510  167326           CMP     #STACK0,$SP     ;CHECK THE STACK POINTER (R0)
2300  011106  001402                           BEQ     .+6             ;BRANCH IF OK
2301  011110  104000                           HLT                     ;STACK POINTER (R0) NOT EQUAL TO #STACK0
2302  011112  000354                           354                     ;THE ERROR NUMBER IS 354
2303
2304  011114  022767  011036  167314           CMP     #RTA51, ANS1    ;CHECK FIS TRAP RETURN ADDRESS
2305  011122  001402                           BEQ     .+6             ;BRANCH IF OK
2306  011124  104001                           HLT+1                   ;FIS TRAP AT WRONG ADDRESS
2307  011126  000355                           355                     ;THE ERROR NUMBER IS 355
2308
2309  011130  022767  000212  167302           CMP     #212,   ANS2    ;CHECK PS BEFORE FIS TRAP
2310  011136  001402                           BEQ     .+6             ;BRANCH IF OK
2311  011140  104002                           HLT+2                   ;PS AT FIS TRAP TIME NOT 212
2312  011142  000356                           356                     ;THE ERROR NUMBER IS 356
2313
2314  011144  022767  114100  167270           CMP     #114100,ANS3    ;CHECK DATA FROM THE STACK
2315  011152  001402                           BEQ     .+6             ;BRANCH IF OK
2316  011154  104004                           HLT+4                   ;DATA ON STACK (114100) CHANGED
2317  011156  000357                           357                     ;THE ERROR NUMBER IS 357
2318
2319  011160  005767  167260                   TST     ANS4            ;CHECK DATA FROM STACK
2320  011164  001402                           BEQ     .+6             ;BRANCH IF OK
2321  011166  104004                           HLT+4                   ;DATA ON STACK (000000) CHANGED
2322  011170  000360                           360                     ;THE ERROR NUMBER IS 360
2323
2324  011172  022767  024252  167246           CMP     #024252,ANS5    ;CHECK DATA FROM STACK
2325  011200  001402                           BEQ     .+6             ;BRANCH IF OK
```

```
2326  011202  104006                        HLT+6                   ;DATA ON STACK (024252) CHANGED
2327  011204  000361                        361                     ;THE ERROR NUMBER IS 361
2328
2329  011206  022767  125252  167234        CMP     #125252,ANS6    ;CHECK DATA FROM STACK
2330  011214  001402                        BEQ     .+6             ;BRANCH IF OK
2331  011216  104006                        HLT+6                   ;DATA ON STACK (125252) CHANGED
2332  011220  000362                        362                     ;THE ERROR NUMBER IS 362
2333
2334  011222  122767  000051  167154  END51: CMPB    #51,    $TESTN  ;CHECK THE TEST NUMBER
2335  011230  001402                        BEQ     .+6             ;BRANCH IF OK
2336  011232  104000                        HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
2337  011234  000363                        363                     ;THE ERROR NUMBER IS 363
2338
2339
```

```
2340
2341                            ;*************************************************************************
2342                            ;TEST 52:       FMUL (LSI-11 FLOATING MULTIPLY INSTRUCTION)
2343                            ;       076452,125252 * 041500,000001 ==> OVERFLOW
2344                            ;       PS(ON STACK) = 002,     STACK POINTER = SP
2345                            ;*************************************************************************
2346
2347 011236  104400                           SCOPE
2348 011240  004567  004554     TST52:  JSR     R5,     PUSHS   ;PUSH 4 WORDS ONTO STACK, SET PRIORITY
2349 011244  041500  000001             .WORD   041500,000001   ;SECOND OPERAND ON TOP
2350 011250  076452  125252             .WORD   076452,125252   ;FIRST OPERAND ON BOTTOM
2351 011254  000105                     .WORD   105             ;PROCESSOR PRIORITY LEVEL
2352 011256  011304  000357             .WORD   ISR52,  357     ;FIS TRAP VECTOR
2353
2354 011262  000240                     NOP
2355 011264  075026                     FMUL    SP              ;FLOATING MULTIPLY ON THE STACK
2356
2357 011266  004767  004566     RTA52:  JSR     PC,     POPS    ;POP THE "ANSWER"
2358 011272  104002                     HLT+2                   ;FIS TRAP DIDN'T OCCURE!
2359 011274  000364                     364                     ;THE ERROR NUMBER IS 364
2360 011276  012706  000600             MOV     #BEGIN, SP      ;RESTORE THE STACK POINTER
2361 011302  000464                     BR      END52
2362
2363 011304  004767  004602     ISR52:  JSR     PC,     POPES   ;POP ALL DATA OFF THE STACK
2364 011310  022706  000600             CMP     #BEGIN, SP      ;CHECK THE STACK POINTER
2365 011314  001405                     BEQ     ISA52           ;BRANCH IF OK
2366 011316  012706  000600             MOV     #BEGIN, SP      ;RESTORE THE STACK POINTER
2367 011322  104000                     HLT                     ;STACK POINTER FOULED UP
2368 011324  000365                     365                     ;THE ERROR NUMBER IS 365
2369 011326  000452                     BR      END52           ;SKIP REST OF TEST
2370
2371 011330  122767  000357  167074 ISA52:  CMPB    #357,   $PSW    ;CHECK PS AFTER FIS TRAP
2372 011336  001402                     BEQ     .+6             ;BRANCH IF OK
2373 011340  104000                     HLT                     ;PS AFTER FIS TRAP NOT EQUAL TO 357
2374 011342  000366                     366                     ;THE ERROR NUMBER IS 366
2375
2376 011344  022767  011266  167064             CMP     #RTA52, ANS1    ;CHECK FIS TRAP RETURN ADDRESS
2377 011352  001402                     BEQ     .+6             ;BRANCH IF OK
2378 011354  104001                     HLT+1                   ;FIS TRAP AT WRONG ADDRESS
2379 011356  000367                     367                     ;THE ERROR NUMBER IS 367
2380
2381 011360  022767  000002  167052             CMP     #002,   ANS2    ;CHECK PS BEFORE FIS TRAP
2382 011366  001402                     BEQ     .+6             ;BRANCH IF OK
2383 011370  104002                     HLT+2                   ;PS AT FIS TRAP TIME NOT 002
2384 011372  000370                     370                     ;THE ERROR NUMBER IS 370
2385
2386 011374  022767  041500  167040             CMP     #041500,ANS3    ;CHECK DATA FROM THE STACK
2387 011402  001402                     BEQ     .+6             ;BRANCH IF OK
2388 011404  104004                     HLT+4                   ;DATA ON STACK (041500) CHANGED
2389 011406  000371                     371                     ;THE ERROR NUMBER IS 371
2390
2391 011410  022767  000001  167026             CMP     #000001,ANS4    ;CHECK DATA FROM STACK
2392 011416  001402                     BEQ     .+6             ;BRANCH IF OK
2393 011420  104004                     HLT+4                   ;DATA ON STACK (000001) CHANGED
2394 011422  000372                     372                     ;THE ERROR NUMBER IS 372
2395
```

```
2396  011424  022767  076452  167014          CMP     #076452,ANS5    ;CHECK DATA FROM STACK
2397  011432  001402                          BEQ     .+6             ;BRANCH IF OK
2398  011434  104006                          HLT+6                   ;DATA ON STACK (076452) CHANGED
2399  011436  000373                          373                     ;THE ERROR NUMBER IS 373
2400
2401  011440  022767  125252  167002          CMP     #125252,ANS6    ;CHECK DATA FROM STACK
2402  011446  001402                          BEQ     .+6             ;BRANCH IF OK
2403  011450  104006                          HLT+6                   ;DATA ON STACK (125252) CHANGED
2404  011452  000374                          374                     ;THE ERROR NUMBER IS 374
2405
2406  011454  122767  000052  166722  END52:  CMPB    #52,    $TESTN  ;CHECK THE TEST NUMBER
2407  011462  001402                          BEQ     .+6             ;BRANCH IF OK
2408  011464  104000                          HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
2409  011466  000375                          375                     ;THE ERROR NUMBER IS 375
2410
2411
```

```
2412
2413                                    ;*************************************************************************
2414                                    ;TEST 53:       FDIV (LSI-11 FLOATING DIVIDE INSTRUCTION)
2415                                    ;       167452,125251 / 127652,125252 = 077777,177776
2416                                    ;       PS = 000,       STACK POINTER = R0
2417                                    ;*************************************************************************
2418
2419  011470  104400                            SCOPE
2420  011472  004567  004474           TST53:  JSR     R5,     PUSHR   ;PUSH 4 WORDS ONTO R0 STACK, SET PRIORITY
2421  011476  127652  125252                   .WORD   127652,125252   ;SECOND OPERAND ON TOP
2422  011502  167452  125251                   .WORD   167452,125251   ;FIRST OPERAND ON BOTTOM
2423  011506  000111                            .WORD   111             ;PROCESSOR PRIORITY LEVEL
2424  011510  016442  000340                   .WORD   TRAPER, 340     ;FIS TRAP VECTOR
2425  011514  012700  000510                   MOV     #STACK0,R0      ;SHECK  STACK POINTER
2426
2427  011520  000240                            NOP
2428  011522  075030                            FDIV    R0              ;FLOATING DIVIDE ON THE R0 STACK
2429
2430  011524  004767  004474                   JSR     PC,     POPR    ;POP THE ANSWER
2431  011530  010067  166700                   MOV     R0,     $SP     ;SAVE "STACK POINTER"
2432  011534  105767  166672                   TSTB    $PSW            ;CHECK PS (EXCEPT T BIT)
2433  011540  001402                            BEQ     .+6             ;BRANCH IF OK
2434  011542  104000                            HLT                     ;PS NOT EQUAL TO 000
2435  011544  000376                            376                     ;THE ERROR NUMBER IS 376
2436
2437  011546  022767  000514  166660           CMP     #STACK4,$SP     ;CHECK THE STACK POINTER (R0)
2438  011554  001402                            BEQ     .+6             ;BRANCH IF OK
2439  011556  104000                            HLT                     ;STACK POINTER (R0) NOT EQUAL TO #STACK4
2440  011560  000377                            377                     ;THE ERROR NUMBER IS 377
2441
2442  011562  022767  077777  166646           CMP     #077777,ANS1    ;CHECK FIRST HALF OF ANSWER
2443  011570  001402                            BEQ     .+6             ;BRANCH IF OK
2444  011572  104002                            HLT+2                   ;ANS1 NOT EQUAL TO 077777
2445  011574  000400                            400                     ;THE ERROR NUMBER IS 400
2446
2447  011576  022767  177776  166634           CMP     #177776,ANS2    ;CHECK SECOND HALF OF ANSWER
2448  011604  001402                            BEQ     .+6             ;BRANCH IF OK
2449  011606  104002                            HLT+2                   ;ANS2 NOT EQUAL TO 177776
2450  011610  000401                            401                     ;THE ERROR NUMBER IS 401
2451
2452  011612  122767  000053  166564  END53:   CMPB    #53,    $TESTN  ;CHECK THE TEST NUMBER
2453  011620  001402                            BEQ     .+6             ;BRANCH IF OK
2454  011622  104000                            HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
2455  011624  000402                            402                     ;THE ERROR NUMBER IS 402
2456
2457
```

```
                                ;*****************************************************************************
                                ;TEST 54:       FDIV (LSI-11 FLOATING DIVIDE INSTRUCTION)
                                ;       167452,125252 / 027652,125253 = 177777,177777
                                ;       PS = 210,       STACK POINTER = SP
                                ;*****************************************************************************

                                        SCOPE
011626  104400
011630  004567  004164          TST54:  JSR     R5,     PUSHS   ;PUSH 4 WORDS ONTO STACK, SET PRIORITY
011634  027652  125253                  .WORD   027652,125253   ;SECOND OPERAND ON TOP
011640  167452  125252                  .WORD   167452,125252   ;FIRST OPERAND ON BOTTOM
011644  000300                          .WORD   300             ;PROCESSOR PRIORITY LEVEL
011646  016442  000340                  .WORD   TRAPER, 340     ;FIS TRAP VECTOR

011652  000240                          NOP
011654  075036                          FDIV    SP              ;FLOATING DIVIDE ON THE STACK

011656  004767  004176                  JSR     PC,     POPS    ;POP THE ANSWER
011662  022706  000600                  CMP     #BEGIN, SP      ;CHECK THE STACK POINTER
011666  001405                          BEQ     TSA54           ;BRANCH IF OK
011670  012706  000600                  MOV     #BEGIN, SP      ;RESTORE STACK POINTER
011674  104000                          HLT                     ;STACK POINTER FOULED UP
011676  000403                          403                     ;THE ERROR NUMBER IS 403
011700  000422                          BR      END54           ;SKIP REST OF TEST

011702  122767  000210  166522  TSA54:  CMPB    #210,   $PSW    ;CHECK PS (EXCEPT T BIT)
011710  001402                          BEQ     .+6             ;BRANCH IF OK
011712  104000                          HLT                     ;PS NOT EQUAL TO 210
011714  000404                          404                     ;THE ERROR NUMBER IS 404

011716  022767  177777  166512          CMP     #177777,ANS1    ;CHECK FIRST HALF OF ANSWER
011724  001402                          BEQ     .+6             ;BRANCH IF OK
011726  104002                          HLT+2                   ;ANS1 NOT EQUAL TO 177777
011730  000405                          405                     ;THE ERROR NUMBER IS 405

011732  022767  177777  166500          CMP     #177777,ANS2    ;CHECK SECOND HALF OF ANSWER
011740  001402                          BEQ     .+6             ;BRANCH IF OK
011742  104002                          HLT+2                   ;ANS2 NOT EQUAL TO 177777
011744  000406                          406                     ;THE ERROR NUMBER IS 406

011746  122767  000054  166430  END54:  CMPB    #54,    $TESTN  ;CHECK THE TEST NUMBER
011754  001402                          BEQ     .+6             ;BRANCH IF OK
011756  104000                          HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
011760  000407                          407                     ;THE ERROR NUMBER IS 407
```

```
                                ;*****************************************************************************
                                ;TEST 55:       FDIV (LSI-11 FLOATING DIVIDE INSTRUCTION)
                                ;       125252,125252 / 065252,125252 = 100200,000000
                                ;       PS = 210,       STACK POINTER = R2
                                ;*****************************************************************************

011762  104400                          SCOPE
011764  004567  004202          TST55:  JSR     R5,     PUSHR   ;PUSH 4 WORDS ONTO R2 STACK, SET PRIORITY
011770  065252  125252                  .WORD   065252,125252   ;SECOND OPERAND ON TOP
011774  125252  125252                  .WORD   125252,125252   ;FIRST OPERAND ON BOTTOM
012000  000217                          .WORD   217             ;PROCESSOR PRIORITY LEVEL
012002  016442  000340                  .WORD   TRAPER, 340     ;FIS TRAP VECTOR
012006  012702  000510                  MOV     #STACK0,R2      ;SHECK  STACK POINTER

012012  000240                          NOP
012014  075032                          FDIV    R2              ;FLOATING DIVIDE ON THE R2 STACK

012016  004767  004202                  JSR     PC,     POPR    ;POP THE ANSWER
012022  010267  166406                  MOV     R2,     $SP     ;SAVE "STACK POINTER"
012026  122767  000210  166376          CMPB    #210,   $PSW    ;CHECK PS (EXCEPT T BIT)
012034  001402                          BEQ     .+6             ;BRANCH IF OK
012036  104000                          HLT                     ;PS NOT EQUAL TO 210
012040  000410                          410                     ;THE ERROR NUMBER IS 410

012042  022767  000514  166364          CMP     #STACK4,$SP     ;CHECK THE STACK POINTER (R2)
012050  001402                          BEQ     .+6             ;BRANCH IF OK
012052  104000                          HLT                     ;STACK POINTER (R2) NOT EQUAL TO #STACK4
012054  000411                          411                     ;THE ERROR NUMBER IS 411

012056  022767  100200  166352          CMP     #100200,ANS1    ;CHECK FIRST HALF OF ANSWER
012064  001402                          BEQ     .+6             ;BRANCH IF OK
012066  104002                          HLT+2                   ;ANS1 NOT EQUAL TO 100200
012070  000412                          412                     ;THE ERROR NUMBER IS 412

012072  005767  166342                  TST     ANS2            ;CHECK SECOND HALF OF ANSWER
012076  001402                          BEQ     .+6             ;BRANCH IF OK
012100  104002                          HLT+2                   ;ANS2 NOT EQUAL TO 000000
012102  000413                          413                     ;THE ERROR NUMBER IS 413

012104  122767  000055  166272  END55:  CMPB    #55,    $TESTN  ;CHECK THE TEST NUMBER
012112  001402                          BEQ     .+6             ;BRANCH IF OK
012114  104000                          HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
012116  000414                          414                     ;THE ERROR NUMBER IS 414
```

```
                                ;*************************************************************************
                                ;TEST 56:       FDIV (LSI-11 FLOATING DIVIDE INSTRUCTION)
                                ;       000000,000000 / 140670,123456 = 000000,000000
                                ;       PS = 004,       STACK POINTER = R3
                                ;*************************************************************************

                                        SCOPE
012120  104400                  
012122  004567  004044          TST56:  JSR     R5,     PUSHR   ;PUSH 4 WORDS ONTO R3 STACK, SET PRIORITY
012126  140670  123456                  .WORD   140670,123456   ;SECOND OPERAND ON TOP
012132  000000  000000                  .WORD   000000,000000   ;FIRST OPERAND ON BOTTOM
012136  000105                          .WORD   105             ;PROCESSOR PRIORITY LEVEL
012140  016442  000340                  .WORD   TRAPER, 340     ;FIS TRAP VECTOR
012144  012703  000510                  MOV     #STACK0,R3      ;SHECK  STACK POINTER

012150  000240                          NOP
012152  075033                          FDIV    R3              ;FLOATING DIVIDE ON THE R3 STACK

012154  004767  004044                  JSR     PC,     POPR    ;POP THE ANSWER
012160  010367  166250                  MOV     R3,     $SP     ;SAVE "STACK POINTER"
012164  122767  000004  166240          CMPB    #004,   $PSW    ;CHECK PS (EXCEPT T BIT)
012172  001402                          BEQ     .+6             ;BRANCH IF OK
012174  104000                          HLT                     ;PS NOT EQUAL TO 004
012176  000415                          415                     ;THE ERROR NUMBER IS 415

012200  022767  000514  166226          CMP     #STACK4,$SP     ;CHECK THE STACK POINTER (R3)
012206  001402                          BEQ     .+6             ;BRANCH IF OK
012210  104000                          HLT                     ;STACK POINTER (R3) NOT EQUAL TO #STACK4
012212  000416                          416                     ;THE ERROR NUMBER IS 416

012214  005767  166216                  TST     ANS1            ;CHECK FIRST HALF OF ANSWER
012220  001402                          BEQ     .+6             ;BRANCH IF OK
012222  104002                          HLT+2                   ;ANS1 NOT EQUAL TO 000000
012224  000417                          417                     ;THE ERROR NUMBER IS 417

012226  005767  166206                  TST     ANS2            ;CHECK SECOND HALF OF ANSWER
012232  001402                          BEQ     .+6             ;BRANCH IF OK
012234  104002                          HLT+2                   ;ANS2 NOT EQUAL TO 000000
012236  000420                          420                     ;THE ERROR NUMBER IS 420

012240  122767  000056  166136  END56:  CMPB    #56,    $TESTN  ;CHECK THE TEST NUMBER
012246  001402                          BEQ     .+6             ;BRANCH IF OK
012250  104000                          HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
012252  000421                          421                     ;THE ERROR NUMBER IS 421
```

```
                                ;*************************************************************************
                                ;TEST 57:       FDIV (LSI-11 FLOATING DIVIDE INSTRUCTION)
                                ;       102500,146000 / 104000,104000 = 036700,000000
                                ;       PS = 200,       STACK POINTER = PC
                                ;*************************************************************************

012254  104400                          SCOPE
012256  004567  004062          TST57:  JSR     R5,     PUSH7   ;PUSH 4 WORDS ONTO STACK, SET PRIORITY
012262  012306                          .WORD   STK57           ;TOP OF STACK
012264  104000  104000                  .WORD   104000,104000   ;SECOND OPERAND ON TOP
012270  102500  146000                  .WORD   102500,146000   ;FIRST OPERAND ON BOTTOM
012274  000357                          .WORD   357             ;PROCESSOR PRIORITY LEVEL
012276  016442  000340                  .WORD   TRAPER,340      ;FIS TRAP VECTOR

012302  000240                          NOP
012304  075037                          FDIV    PC              ;FLOATING DIVIDE ON FOLLOWING 4 WORDS
012306  104000          STK57:  104000  ;SHOULD CONTAIN 104000
012310  104000                  104000  ;SHOULD CONTAIN 104000
012312  102500                  102500  ;BEFORE FDIV, 102500; AFTER, 036700
012314  146000                  146000  ;BEFORE FDIV, 146000; AFTER, 000000

012316  004767  004052                  JSR     PC,     POP7    ;POP THE ANSWER
012322  122767  000200  166102          CMPB    #200,   $PSW    ;CHECK PS (EXCEPT T BIT)
012330  001402                          BEQ     .+6             ;BRANCH IF OK
012332  104000                          HLT                     ;PS NOT EQUAL TO 200
012334  000422                          422                     ;THE ERROR NUMBER IS 422

012336  022767  104000  166072          CMP     #104000,ANS1    ;CHECK FIRST HALF OF INPUT DATA (STK57)
012344  001402                          BEQ     .+6             ;BRANCH IF OK
012346  104002                          HLT+2                   ;ANS1 NOT EQUAL TO 104000
012350  000423                          423                     ;THE ERROR NUMBER IS 423

012352  022767  104000  166060          CMP     #104000,ANS2    ;CHECK SECOND HALF OF INPUT DATA (STK57+2)
012360  001402                          BEQ     .+6             ;BRANCH IF OK
012362  104002                          HLT+2                   ;ANS2 NOT EQUAL TO 104000
012364  000424                          424                     ;THE ERROR NUMBER IS 424

012366  022767  036700  166046          CMP     #036700,ANS3    ;CHECK FIRST HALF OF ANSWER
012374  001402                          BEQ     .+6             ;BRANCH IF OK
012376  104004                          HLT+4                   ;ANS3 NOT EQUAL TO 036700
012400  000425                          425                     ;THE ERROR NUMBER IS 425

012402  005767  166036                  TST     ANS4            ;CHECK SECOND HALF OF ANSWER
012406  001402                          BEQ     .+6             ;BRANCH IF OK
012410  104004                          HLT+4                   ;ANS4 NOT EQUAL TO 000000
012412  000426                          426                     ;THE ERROR NUMBER IS 426

012414  122767  000057  165762  END57:  CMPB    #57,    $TESTN  ;CHECK THE TEST NUMBER
012422  001402                          BEQ     .+6             ;BRANCH IF OK
012424  104000                          HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
012426  000427                          427                     ;THE ERROR NUMBER IS 427


                                ;*************************************************************************
```

```
                                ;TEST 60:       FDIV (LSI-11 FLOATING DIVIDE INSTRUCTION)
                                ;       025252,125251 / 065252,125252 ==> UNDERFLOW
                                ;       PS(ON STACK) = 012,     STACK POINTER = R1
                                ;*****************************************************************************

012430  104400                          SCOPE
012432  004567  003534          TST60:  JSR     R5,     PUSHR   ;PUSH 4 WORDS ONTO R1 STACK, SET PRIORITY
012436  065252  125252                  .WORD   065252,125252   ;SECOND OPERAND ON TOP
012442  025252  125251                  .WORD   025252,125251   ;FIRST OPERAND ON BOTTOM
012446  000015                          .WORD   015             ;PROCESSOR PRIORITY LEVEL
012450  012502  000300                  .WORD   ISR60,  300     ;FIS TRAP VECTOR
012454  012701  000510                  MOV     #STACK0,R1      ;SET UP R1 AS STACK POINTER

012460  000240                          NOP
012462  075031                          FDIV    R1              ;FLOATING DIVIDE ON THE R1 STACK

012464  004767  003534          RTA60:  JSR     PC,     POPR    ;POP THE "ANSWER"
012470  010167  165740                  MOV     R1,     $SP     ;SAVE STACK POINTER (R1)
012474  104002                          HLT+2                   ;FIS TRAP DIDN'T OCCURE!
012476  000430                          430                     ;THE ERROR NUMBER IS 430
012500  000464                          BR      END60

012502  004767  003546          ISR60:  JSR     PC,     POPER   ;POP ALL DATA OFF THE STACKS
012506  010167  165722                  MOV     R1,     $SP     ;SAVE STACK POINTER (R1)
012512  122767  000300  165712          CMPB    #300,   $PSW    ;CHECK PS AFTER FIS TRAP
012520  001402                          BEQ     .+6             ;BRANCH IF OK
012522  104000                          HLT                     ;PS AFTER FIS TRAP NOT EQUAL TO 300
012524  000431                          431                     ;THE ERROR NUMBER IS 431

012526  022767  000510  165700          CMP     #STACK0,$SP     ;CHECK THE STACK POINTER (R1)
012534  001402                          BEQ     .+6             ;BRANCH IF OK
012536  104000                          HLT                     ;STACK POINTER (R1) NOT EQUAL TO #STACK0
012540  000432                          432                     ;THE ERROR NUMBER IS 432

012542  022767  012464  165666          CMP     #RTA60, ANS1    ;CHECK FIS TRAP RETURN ADDRESS
012550  001402                          BEQ     .+6             ;BRANCH IF OK
012552  104001                          HLT+1                   ;FIS TRAP AT WRONG ADDRESS
012554  000433                          433                     ;THE ERROR NUMBER IS 433

012556  022767  000012  165654          CMP     #012,   ANS2    ;CHECK PS BEFORE FIS TRAP
012564  001402                          BEQ     .+6             ;BRANCH IF OK
012566  104002                          HLT+2                   ;PS AT FIS TRAP TIME NOT 012
012570  000434                          434                     ;THE ERROR NUMBER IS 434

012572  022767  065252  165642          CMP     #065252,ANS3    ;CHECK DATA FROM THE STACK
012600  001402                          BEQ     .+6             ;BRANCH IF OK
012602  104004                          HLT+4                   ;DATA ON STACK (065252) CHANGED
012604  000435                          435                     ;THE ERROR NUMBER IS 435

012606  022767  125252  165630          CMP     #125252,ANS4    ;CHECK DATA FROM STACK
012614  001402                          BEQ     .+6             ;BRANCH IF OK
012616  104004                          HLT+4                   ;DATA ON STACK (125252) CHANGED
012620  000436                          436                     ;THE ERROR NUMBER IS 436

012622  022767  025252  165616          CMP     #025252,ANS5    ;CHECK DATA FROM STACK
012630  001402                          BEQ     .+6             ;BRANCH IF OK
```

```
2708  012632  104006                          HLT+6                   ;DATA ON STACK (025252) CHANGED
2709  012634  000437                          437                     ;THE ERROR NUMBER IS 437
2710
2711  012636  022767  125251  165604          CMP     #125251,ANS6    ;CHECK DATA FROM STACK
2712  012644  001402                          BEQ     .+6             ;BRANCH IF OK
2713  012646  104006                          HLT+6                   ;DATA ON STACK (125251) CHANGED
2714  012650  000440                          440                     ;THE ERROR NUMBER IS 440
2715
2716  012652  122767  000060  165524  END60:  CMPB    #60,    $TESTN  ;CHECK THE TEST NUMBER
2717  012660  001402                          BEQ     .+6             ;BRANCH IF OK
2718  012662  104000                          HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
2719  012664  000441                          441                     ;THE ERROR NUMBER IS 441
2720
2721
```

```
                                ;*************************************************************************
                                ;TEST 61:       FDIV (LSI-11 FLOATING DIVIDE INSTRUCTION)
                                ;       067452,125252 / 127652,125252 ==> OVERFLOW
                                ;       PS(ON STACK) = 202,     STACK POINTER = R4
                                ;*************************************************************************

012666  104400                          SCOPE
012670  004567  003276          TST61:  JSR     R5,     PUSHR   ;PUSH 4 WORDS ONTO R4 STACK, SET PRIORITY
012674  127652  125252                  .WORD   127652,125252   ;SECOND OPERAND ON TOP
012700  067452  125252                  .WORD   067452,125252   ;FIRST OPERAND ON BOTTOM
012704  000242                          .WORD   242             ;PROCESSOR PRIORITY LEVEL
012706  012740  000357                  .WORD   ISR61,  357     ;FIS TRAP VECTOR
012712  012704  000510                  MOV     #STACKO,R4      ;SET UP R4 AS STACK POINTER

012716  000240                          NOP
012720  075034                          FDIV    R4              ;FLOATING DIVIDE ON THE R4 STACK

012722  004767  003276          RTA61:  JSR     PC,     POPR    ;POP THE "ANSWER"
012726  010467  165502                  MOV     R4,     $SP     ;SAVE STACK POINTER (R4)
012732  104002                          HLT+2                   ;FIS TRAP DIDN'T OCCURE!
012734  000442                          442                     ;THE ERROR NUMBER IS 442
012736  000464                          BR      END61

012740  004767  003310          ISR61:  JSR     PC,     POPER   ;POP ALL DATA OFF THE STACKS
012744  010467  165464                  MOV     R4,     $SP     ;SAVE STACK POINTER (R4)
012750  122767  000357  165454          CMPB    #357,   $PSW    ;CHECK PS AFTER FIS TRAP
012756  001402                          BEQ     .+6             ;BRANCH IF OK
012760  104000                          HLT                     ;PS AFTER FIS TRAP NOT EQUAL TO 357
012762  000443                          443                     ;THE ERROR NUMBER IS 443

012764  022767  000510  165442          CMP     #STACKO,$SP     ;CHECK THE STACK POINTER (R4)
012772  001402                          BEQ     .+6             ;BRANCH IF OK
012774  104000                          HLT                     ;STACK POINTER (R4) NOT EQUAL TO #STACKO
012776  000444                          444                     ;THE ERROR NUMBER IS 444

013000  022767  012722  165430          CMP     #RTA61, ANS1    ;CHECK FIS TRAP RETURN ADDRESS
013006  001402                          BEQ     .+6             ;BRANCH IF OK
013010  104001                          HLT+1                   ;FIS TRAP AT WRONG ADDRESS
013012  000445                          445                     ;THE ERROR NUMBER IS 445

013014  022767  000202  165416          CMP     #202,   ANS2    ;CHECK PS BEFORE FIS TRAP
013022  001402                          BEQ     .+6             ;BRANCH IF OK
013024  104002                          HLT+2                   ;PS AT FIS TRAP TIME NOT 202
013026  000446                          446                     ;THE ERROR NUMBER IS 446

013030  022767  127652  165404          CMP     #127652,ANS3    ;CHECK DATA FROM THE STACK
013036  001402                          BEQ     .+6             ;BRANCH IF OK
013040  104004                          HLT+4                   ;DATA ON STACK (127652) CHANGED
013042  000447                          447                     ;THE ERROR NUMBER IS 447

013044  022767  125252  165372          CMP     #125252,ANS4    ;CHECK DATA FROM STACK
013052  001402                          BEQ     .+6             ;BRANCH IF OK
013054  104004                          HLT+4                   ;DATA ON STACK (125252) CHANGED
013056  000450                          450                     ;THE ERROR NUMBER IS 450
```

```
 2778  013060  022767  067452  165360          CMP     #067452,ANS5    ;CHECK DATA FROM STACK
 2779  013066  001402                          BEQ     .+6             ;BRANCH IF OK
 2780  013070  104006                          HLT+6                   ;DATA ON STACK (067452) CHANGED
 2781  013072  000451                          451                     ;THE ERROR NUMBER IS 451
 2782
 2783  013074  022767  125252  165346          CMP     #125252,ANS6    ;CHECK DATA FROM STACK
 2784  013102  001402                          BEQ     .+6             ;BRANCH IF OK
 2785  013104  104006                          HLT+6                   ;DATA ON STACK (125252) CHANGED
 2786  013106  000452                          452                     ;THE ERROR NUMBER IS 452
 2787
 2788  013110  122767  000061  165266  END61:  CMPB    #61,    $TESTN  ;CHECK THE TEST NUMBER
 2789  013116  001402                          BEQ     .+6             ;BRANCH IF OK
 2790  013120  104000                          HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
 2791  013122  000453                          453                     ;THE ERROR NUMBER IS 453
 2792
 2793
```

```
2794
2795                            ;*************************************************************************
2796                            ;TEST 62:       FDIV (LSI-11 FLOATING DIVIDE INSTRUCTION)
2797                            ;       052525,052525 / 100125,125252 ==> DIVIDE BY ZERO
2798                            ;       PS(ON STACK) = 013,     STACK POINTER = R5
2799                            ;*************************************************************************
2800
2801  013124  104400                       SCOPE
2802  013126  004567  003040       TST62:  JSR     R5,     PUSHR   ;PUSH 4 WORDS ONTO R5 STACK, SET PRIORITY
2803  013132  100125  125252               .WORD   100125,125252   ;SECOND OPERAND ON TOP
2804  013136  052525  052525               .WORD   052525,052525   ;FIRST OPERAND ON BOTTOM
2805  013142  000047                       .WORD   047             ;PROCESSOR PRIORITY LEVEL
2806  013144  013176  000113               .WORD   ISR62,  113     ;FIS TRAP VECTOR
2807  013150  012705  000510               MOV     #STACK0,R5      ;SET UP R5 AS STACK POINTER
2808
2809  013154  000240                       NOP
2810  013156  075035                       FDIV    R5              ;FLOATING DIVIDE ON THE R5 STACK
2811
2812  013160  004767  003040       RTA62:  JSR     PC,     POPR    ;POP THE "ANSWER"
2813  013164  010567  165244               MOV     R5,     $SP     ;SAVE STACK POINTER (R5)
2814  013170  104002                       HLT+2                   ;FIS TRAP DIDN'T OCCURE!
2815  013172  000454                       454                     ;THE ERROR NUMBER IS 454
2816  013174  000464                       BR      END62
2817
2818  013176  004767  003052       ISR62:  JSR     PC,     POPER   ;POP ALL DATA OFF THE STACKS
2819  013202  010567  165226               MOV     R5,     $SP     ;SAVE STACK POINTER (R5)
2820  013206  122767  000113  165216       CMPB    #113,   $PSW    ;CHECK PS AFTER FIS TRAP
2821  013214  001402                       BEQ     .+6             ;BRANCH IF OK
2822  013216  104000                       HLT                     ;PS AFTER FIS TRAP NOT EQUAL TO 113
2823  013220  000455                       455                     ;THE ERROR NUMBER IS 455
2824
2825  013222  022767  000510  165204       CMP     #STACK0,$SP     ;CHECK THE STACK POINTER (R5)
2826  013230  001402                       BEQ     .+6             ;BRANCH IF OK
2827  013232  104000                       HLT                     ;STACK POINTER (R5) NOT EQUAL TO #STACK0
2828  013234  000456                       456                     ;THE ERROR NUMBER IS 456
2829
2830  013236  022767  013160  165172       CMP     #RTA62, ANS1    ;CHECK FIS TRAP RETURN ADDRESS
2831  013244  001402                       BEQ     .+6             ;BRANCH IF OK
2832  013246  104001                       HLT+1                   ;FIS TRAP AT WRONG ADDRESS
2833  013250  000457                       457                     ;THE ERROR NUMBER IS 457
2834
2835  013252  022767  000013  165160       CMP     #013,   ANS2    ;CHECK PS BEFORE FIS TRAP
2836  013260  001402                       BEQ     .+6             ;BRANCH IF OK
2837  013262  104002                       HLT+2                   ;PS AT FIS TRAP TIME NOT 013
2838  013264  000460                       460                     ;THE ERROR NUMBER IS 460
2839
2840  013266  022767  100125  165146       CMP     #100125,ANS3    ;CHECK DATA FROM THE STACK
2841  013274  001402                       BEQ     .+6             ;BRANCH IF OK
2842  013276  104004                       HLT+4                   ;DATA ON STACK (100125) CHANGED
2843  013300  000461                       461                     ;THE ERROR NUMBER IS 461
2844
2845  013302  022767  125252  165134       CMP     #125252,ANS4    ;CHECK DATA FROM STACK
2846  013310  001402                       BEQ     .+6             ;BRANCH IF OK
2847  013312  104004                       HLT+4                   ;DATA ON STACK (125252) CHANGED
2848  013314  000462                       462                     ;THE ERROR NUMBER IS 462
2849
```

```
 2850  013316  022767  052525  165122          CMP     #052525,ANS5    ;CHECK DATA FROM STACK
 2851  013324  001402                          BEQ     .+6             ;BRANCH IF OK
 2852  013326  104006                          HLT+6                   ;DATA ON STACK (052525) CHANGED
 2853  013330  000463                          463                     ;THE ERROR NUMBER IS 463
 2854
 2855  013332  022767  052525  165110          CMP     #052525,ANS6    ;CHECK DATA FROM STACK
 2856  013340  001402                          BEQ     .+6             ;BRANCH IF OK
 2857  013342  104006                          HLT+6                   ;DATA ON STACK (052525) CHANGED
 2858  013344  000464                          464                     ;THE ERROR NUMBER IS 464
 2859
 2860  013346  122767  000062  165030  END62:  CMPB    #62,    $TESTN  ;CHECK THE TEST NUMBER
 2861  013354  001402                          BEQ     .+6             ;BRANCH IF OK
 2862  013356  104000                          HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
 2863  013360  000465                          465                     ;THE ERROR NUMBER IS 465
 2864
 2865
```

```
2866
2867                            ;*****************************************************************************
2868                            ;TEST 63:       FDIV (LSI-11 FLOATING DIVIDE INSTRUCTION)
2869                            ;       100052,052525 / 000006,123456 ==> DIVIDE BY ZERO
2870                            ;       PS(ON STACK) = 213,     STACK POINTER = SP
2871                            ;*****************************************************************************
2872
2873  013362  104400                       SCOPE
2874  013364  004567  002430   TST63:  JSR     R5,     PUSHS   ;PUSH 4 WORDS ONTO STACK, SET PRIORITY
2875  013370  000006  123456           .WORD   000006,123456   ;SECOND OPERAND ON TOP
2876  013374  100052  052525           .WORD   100052,052525   ;FIRST OPERAND ON BOTTOM
2877  013400  000357                   .WORD   357             ;PROCESSOR PRIORITY LEVEL
2878  013402  013430  000311           .WORD   ISR63,  311     ;FIS TRAP VECTOR
2879
2880  013406  000240                   NOP
2881  013410  075036                   FDIV    SP              ;FLOATING DIVIDE ON THE STACK
2882
2883  013412  004767  002442   RTA63:  JSR     PC,     POPS    ;POP THE "ANSWER"
2884  013416  104002                   HLT+2                   ;FIS TRAP DIDN'T OCCURE!
2885  013420  000466                   466                     ;THE ERROR NUMBER IS 466
2886  013422  012706  000600           MOV     #BEGIN, SP      ;RESTORE THE STACK POINTER
2887  013426  000464                   BR      END63
2888
2889  013430  004767  002456   ISR63:  JSR     PC,     POPES   ;POP ALL DATA OFF THE STACK
2890  013434  022706  000600           CMP     #BEGIN, SP      ;CHECK THE STACK POINTER
2891  013440  001405                   BEQ     ISA63           ;BRANCH IF OK
2892  013442  012706  000600           MOV     #BEGIN, SP      ;RESTORE THE STACK POINTER
2893  013446  104000                   HLT                     ;STACK POINTER FOULED UP
2894  013450  000467                   467                     ;THE ERROR NUMBER IS 467
2895  013452  000452                   BR      END63           ;SKIP REST OF TEST
2896
2897  013454  122767  000311  164750 ISA63:  CMPB    #311,   $PSW    ;CHECK PS AFTER FIS TRAP
2898  013462  001402                   BEQ     .+6             ;BRANCH IF OK
2899  013464  104000                   HLT                     ;PS AFTER FIS TRAP NOT EQUAL TO 311
2900  013466  000470                   470                     ;THE ERROR NUMBER IS 470
2901
2902  013470  022767  013412  164740           CMP     #RTA63, ANS1    ;CHECK FIS TRAP RETURN ADDRESS
2903  013476  001402                   BEQ     .+6             ;BRANCH IF OK
2904  013500  104001                   HLT+1                   ;FIS TRAP AT WRONG ADDRESS
2905  013502  000471                   471                     ;THE ERROR NUMBER IS 471
2906
2907  013504  022767  000213  164726           CMP     #213,   ANS2    ;CHECK PS BEFORE FIS TRAP
2908
2909
2910
2911
2912
2913  013512  001402                   BEQ     .+6             ;BRANCH IF OK
2914  013514  104002                   HLT+2                   ;PS AT FIS TRAP TIME NOT 213
2915  013516  000472                   472                     ;THE ERROR NUMBER IS 472
2916
2917  013520  022767  000006  164714           CMP     #000006,ANS3    ;CHECK DATA FROM THE STACK
2918  013526  001402                   BEQ     .+6             ;BRANCH IF OK
2919  013530  104004                   HLT+4                   ;DATA ON STACK (000006) CHANGED
2920  013532  000473                   473                     ;THE ERROR NUMBER IS 473
2921
```

```
2922  013534  022767  123456  164702          CMP     #123456,ANS4    ;CHECK DATA FROM STACK
2923  013542  001402                          BEQ     .+6             ;BRANCH IF OK
2924  013544  104004                          HLT+4                   ;DATA ON STACK (123456) CHANGED
2925  013546  000474                          474                     ;THE ERROR NUMBER IS 474
2926
2927  013550  022767  100052  164670          CMP     #100052,ANS5    ;CHECK DATA FROMONG TK
2928  013556  001402                          BEQ     .+6             ;BRANCH IF OK
2929  013560  104006                          HLT+6                   ;DATA ON STACK (100052) CHANGED
2930  013562  000475                          475                     ;THE ERROR NUMBER IS 475
2931
2932  013564  022767  052525  164656          CMP     #052525,ANS6    ;CHECK DATA FROM STACK
2933  013572  001402                          BEQ     .+6             ;BRANCH IF OK
2934  013574  104006                          HLT+6                   ;DATA ON STACK (052525) CHANGED
2935  013576  000476                          476                     ;THE ERROR NUMBER IS 476
2936
2937  013600  122767  000063  164576  END63:  CMPB    #63,    $TESTN  ;CHECK THE TEST NUMBER
2938  013606  001402                          BEQ     .+6             ;BRANCH IF OK
2939  013610  104000                          HLT                     ;WREST!  PC MUST HAVE FOULED UP.
2940  013612  000477                          477                     ;THE ERROR NUMBER IS 477
2941
2942
```

```
                                ;*****************************************************************************
                                ;TEST 64:       TEST ALL INSTRUCTION TOGETHER
                                ;                           032107,065432 * 045670,123456
                                ;       134343,107070 + -------------------------------  = 137201,115230
                                ;                       (135252,125252 - 040616,016161)
                                ;       PS=010, STACK POINTER=R4
                                ;*****************************************************************************
013614  104400                          SCOPE
013616  012704  000532          TST64:  MOV     #STAK10,R4      ;SET STACK POINTER
013622  012744  107070                  MOV     #107070,-(R4)   ;LOAD DATA ONTO STACK
013626  012744  134343                  MOV     #134343,-(R4)
013632  012744  065432                  MOV     #065432,-(R4)
013636  012744  032107                  MOV     #032107,-(R4)
013642  012744  123456                  MOV     #123456,-(R4)
013646  012744  045670                  MOV     #045670,-(R4)
013652  012744  125252                  MOV     #125252,-(R4)
013656  012744  135252                  MOV     #135252,-(R4)
013662  012744  016161                  MOV     #016161,-(R4)
013666  012744  040616                  MOV     #040616,-(R4)
013672                                  MTPS    #144            ;SET PROCESSOR STATUS
013672  106427                          .WORD   106400!..C
                                
013676  000240                          NOP
013700  075014                          FSUB    R4      ;135252,125252-040616,016161=140616,017434
013702  075034                          FDIV    R4      ;045670,123456/140616,017434=145246,047065
013704  075024                          FMUL    R4      ;032107,065432*145246,047065=137201,106137
013706  075004                          FADD    R4      ;134343,107070+137201,106137=137201,115230
                                
013710                                  MFPS    $PSW            ;SAVE FINAL PS
013710  106767                          .WORD   106700!..C
013714  042767  000020  164510          BIC     #20,    $PSW    ;CLR T-BIT
013722  012467  164510                  MOV     (R4)+,  ANS1    ;SAVE FIRST HALF OF ANSWER
013726  012467  164506                  MOV     (R4)+,  ANS2    ;SAVE SECOND HALF OF ANSWER
013732  010467  164476                  MOV     R4,     $SP     ;SAVE STACK POINTER
013736  122767  000010  164466          CMPB    #010,   $PSW    ;CHECK PS (EXCEPT T BIT)
013744  001402                          BEQ     .+6             ;BRANCH IF OK
013746  104000                          HLT                     ;PS NOT EQUAL TO 010
013750  000500                          500                     ;THE ERROR NUMBER IS 500
                                
013752  022767  000532  164454          CMP     #STAK10,$SP     ;CHECK THE STACK POINTER (R4)
013760  001402                          BEQ     .+6             ;BRANCH IF OK
013762  104000                          HLT                     ;STACK POINTER (R4) NOT EQUAL TO THE
013764  000501                          501                     ;THE ERROR NUMBER IS 501
                                                                ;ADDRESS OF STAK10
                                
013766  022767  137201  164442          CMP     #137201,ANS1    ;CHECK FIRST HALF OF ANSWER
013774  001402                          BEQ     .+6             ;BRANCH IF OK
013776  104002                          HLT+2                   ;ANS1 NOT EQUAL TO 137201
014000  000502                          502                     ;THE ERROR NUMBER IS 502
                                
014002  022767  115230  164430          CMP     #115230,ANS2    ;CHECK SECOND HALF OF ANSWER
014010  001402                          BEQ     .+6             ;BRANCH IF OK
014012  104002                          HLT+2                   ;ANS2 NOT EQUAL TO 115230
014014  000503                          503                     ;THE ERROR NUMBER IS 503
```

```
 2999
 3000  014016  122767  000064  164360  END64:  CMPB    #64,    $TESTN  ;CHECK THE TEST NUMBER
 3001  014024  001402                          BEQ     .+6             ;BRANCH IF OK
 3002  014026  104000                          HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
 3003  014030  000504                          504                     ;THE ERROR NUMBER IS 504
 3004
```

```
3005
3006                            ;*************************************************************************
3007                            ;TEST 65:       TEST THAT STACK POINTER ADDRESS ERROR CAUSES ABORT
3008                            ;       INSTRUCTION = FADD,     STACK POINTER = R2
3009                            ;*************************************************************************
3010
3011  014032  104400                       SCOPE
3012  014034  012737  014124  000004 TST65: MOV     #ISR65, @#4     ;SET UP ADDRESS TRAP VECTOR
3013  014042  012737  000340  000006       MOV     #340,   @#6
3014  014050  004567  002116               JSR     R5,     PUSHR   ;PUSH 4 WORDS ONTO R2 STACK, SET PRIORITY
3015  014054  070707  016161               .WORD   070707,016161   ;SECOND OPERAND ON TOP
3016  014060  146314  143434               .WORD   146314,143434   ;FIRST OPERAND ON BOTTOM
3017  014064  000143                       .WORD   143             ;PROCESSOR PRIORITY LEVEL
3018  014066  016442  000340               .WORD   TRAPER, 340     ;FIS TRAP VECTOR
3019  014072                               MTPS    #143            ;SET PROCESSOR STATUS
3020  014072  106427                       .WORD   106400!..C
3021  014076  012702  177777               MOV     #177777,R2      ;SET UP R2 AS STACK POINTER
3022
3023  014102  000240                       NOP
3024  014104  075002                       FADD    R2              ;FLOATING ADD ON THE R2 STACK
3025
3026  014106  004767  002112       RTA65:  JSR     PC,     POPR    ;POP THE "ANSWER"
3027  014112  010267  164316               MOV     R2,     $SP     ;SAVE STACK POINTER (R2)
3028  014116  104002                       HLT+2                   ;FIS TRAP DIDN'T OCCURE!
3029  014120  000505                       505                     ;THE ERROR NUMBER IS 505
3030  014122  000464                       BR      END65
3031
3032  014124  004767  002124       ISR65:  JSR     PC,     POPER   ;POP ALL DATA OFF THE STACKS
3033  014130  010267  164300               MOV     R2,     $SP     ;SAVE STACK POINTER (R2)
3034  014134  122767  000340  164270       CMPB    #340,   $PSW    ;CHECK PS AFTER ADR. ERR. TRAP
3035  014142  001402                       BEQ     .+6             ;BRANCH IF OK
3036  014144  104000                       HLT                     ;PS AFTER TRAP NOT EQUAL TO 340
3037  014146  000506                       506                     ;THE ERROR NUMBER IS 506
3038
3039  014150  022767  177777  164256       CMP     #177777,$SP     ;CHECK THE STACK POINTER (R2)
3040  014156  001402                       BEQ     .+6             ;BRANCH IF OK
3041  014160  104000                       HLT                     ;STACK POINTER (R2) NOT EQUAL TO #177777
3042  014162  000507                       507                     ;THE ERROR NUMBER IS 507
3043
3044  014164  022767  014106  164244       CMP     #RTA65, ANS1    ;CHECK FIS TRAP RETURN ADDRESS
3045  014172  001402                       BEQ     .+6             ;BRANCH IF OK
3046  014174  104001                       HLT+1                   ;FIS TRAP AT WRONG ADDRESS
3047  014176  000510                       510                     ;THE ERROR NUMBER IS 510
3048
3049  014200  022767  000151  164232       CMP     #151,   ANS2    ;CHECK PS BEFORE FIS TRAP
3050  014206  001402                       BEQ     .+6             ;BRANCH IF OK
3051  014210  104002                       HLT+2                   ;PS AT FIS TRAP TIME NOT 151
3052  014212  000511                       511                     ;THE ERROR NUMBER IS 511
3053
3054  014214  022767  070707  164220       CMP     #070707,ANS3    ;CHECK DATA FROM THE STACK
3055  014222  001402                       BEQ     .+6             ;BRANCH IF OK
3056  014224  104004                       HLT+4                   ;DATA ON STACK (070707) CHANGED
3057  014226  000512                       512                     ;THE ERROR NUMBER IS 512
3058
3059  014230  022767  016161  164206       CMP     #016161,ANS4    ;CHECK DATA FROM STACK
3060  014236  001402                       BEQ     .+6             ;BRANCH IF OK
```

```
 3061  014240  104004                          HLT+4                   ;DATA ON STACK (016161) CHANGED
 3062  014242  000513                          513                     ;THE ERROR NUMBER IS 513
 3063
 3064  014244  022767  146314  164174          CMP     #146314,ANS5    ;CHECK DATA FROM STACK
 3065  014252  001402                          BEQ     .+6             ;BRANCH IF OK
 3066  014254  104006                          HLT+6                   ;DATA ON STACK (146314) CHANGED
 3067  014256  000514                          514                     ;THE ERROR NUMBER IS 514
 3068
 3069  014260  022767  143434  164162          CMP     #143434,ANS6    ;CHECK DATA FROM STACK
 3070  014266  001402                          BEQ     .+6             ;BRANCH IF OK
 3071  014270  104006                          HLT+6                   ;DATA ON STACK (143434) CHANGED
 3072  014272  000515                          515                     ;THE ERROR NUMBER IS 515
 3073  014274  122767  000065  164102  END65:  CMPB    #65,    $TESTN  ;CHECK THE TEST NUMBER
 3074  014302  001402                          BEQ     .+6             ;BRANCH IF OK
 3075  014304  104000                          HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
 3076  014306  000516                          516                     ;THE ERROR NUMBER IS 516
```

```
3077
3078                            ;*************************************************************************
3079                            ;TEST 66:       TEST THAT STACK POINTER ADDRESS ERROR CAUSES ABORT
3080                            ;       INSTRUCTION = FMUL,     STACK POINTER = R5
3081                            ;*************************************************************************
3082
3083  014310  104400                       SCOPE
3084  014312  012737  014402  000004  TST66: MOV     #ISR66, @#4     ;SET UP ADDRESS TRAP VECTOR
3085  014320  012737  000340  000006        MOV     #340,   @#6
3086  014326  004567  001640                JSR     R5,     PUSHR   ;PUSH 4 WORDS ONTO R5 STACK, SET PRIORITY
3087  014332  065432  123456                .WORD   065432,123456   ;SECOND OPERAND ON TOP
3088  014336  037654  032107                .WORD   037654,032107   ;FIRST OPERAND ON BOTTOM
3089  014342  000202                        .WORD   202             ;PROCESSOR PRIORITY LEVEL
3090  014344  016442  000340                .WORD   TRAPER, 340     ;FIS TRAP VECTOR
3091  014350                                MTPS    #202            ;SET PROCESSOR STATUS
3092  014350  106427                        .WORD   106400!..C
3093  014354  012705  160000                MOV     #160000,R5      ;SET UP R5 AS STACK POINTER
3094
3095  014360  000240                        NOP
3096  014362  075025                        FMUL    R5              ;FLOATING MULTIPLY ON THE R5 STACK
3097
3098  014364                        RTA66:  MFPS    $PSW            ;SAVE THE PSW
3099  014364  106767                        .WORD   106700!..C
3100  014370  010567  164040                MOV     R5,     $SP     ;SAVE STACK POINTER (R5)
3101  014374  104000                        HLT                     ;FIS TRAP DIDN'T OCCURE!
3102  014376  000517                        517                     ;THE ERROR NUMBER IS 517
3103  014400  000434                        BR      END66
3104
3105  014402  004767  001646        ISR66:  JSR     PC,     POPER   ;POP ALL DATA OFF THE STACKS
3106  014406  010567  164022                MOV     R5,     $SP     ;SAVE STACK POINTER (R5)
3107  014412  122767  000340  164012        CMPB    #340,   $PSW    ;CHECK PS AFTER ADR. ERR. TRAP
3108  014420  001402                        BEQ     .+6             ;BRANCH IF OK
3109  014422  104000                        HLT                     ;PS AFTER TRAP NOT EQUAL TO 340
3110  014424  000520                        520                     ;THE ERROR NUMBER IS 520
3111
3112  014426  022767  160000  164000        CMP     #160000,$SP     ;CHECK THE STACK POINTER (R5)
3113  014434  001402                        BEQ     .+6             ;BRANCH IF OK
3114  014436  104000                        HLT                     ;STACK POINTER (R5) NOT EQUAL TO #160000
3115  014440  000521                        521                     ;THE ERROR NUMBER IS 521
3116
3117  014442  022767  014364  163766        CMP     #RTA66, ANS1    ;CHECK FIS TRAP RETURN ADDRESS
3118  014450  001402                        BEQ     .+6             ;BRANCH IF OK
3119  014452  104001                        HLT+1                   ;FIS TRAP AT WRONG ADDRESS
3120  014454  000522                        522                     ;THE ERROR NUMBER IS 522
3121
3122  014456  022767  000210  163754        CMP     #210,   ANS2    ;CHECK PS BEFORE FIS TRAP
3123  014464  001402                        BEQ     .+6             ;BRANCH IF OK
3124  014466  104002                        HLT+2                   ;PS AT FIS TRAP TIME NOT 210
3125  014470  000523                        523                     ;THE ERROR NUMBER IS 523
3126
3127  014472  122767  000066  163704  END66: CMPB    #66,    $TESTN  ;CHECK THE TEST NUMBER
3128  014500  001402                        BEQ     .+6             ;BRANCH IF OK
3129  014502  104000                        HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
3130  014504  000524                        524                     ;THE ERROR NUMBER IS 524
```

```
3131  014506  012737  000006  000004          MOV     #6,     @#4     ;RESTORE TIME-OUT VECTOR
3132  014514  005037  000006                  CLR     @#6
3133  014520  012767  000003  164010          MOV     #3,     TIMES   ;REDUCE NUMBER OF ITTERATIONS
3134
3135                          ;*************************************************************************
3136                          ;TEST 67:       TEST THAT FIS ABORTS PROPERLY WHEN INTERUPTED
3137                          ;       035700,143235 + 000177,134543 = 035700,143235
3138                          ;       PS = .PS,       STACK POINTER = R1
3139                          ;*************************************************************************
3140
3141  014526  104400                          SCOPE
3142  014530  132737  000040  000421  TST67:  BITB    #40,@#$ENVM
3143  014536  001174                          BNE     END67+2         ;EXIT THIS TEST IF BIT 5 OF $ENVM IS HIGH
3144  014540  013704  000546                  MOV     @#TTYOUT,R4
3145  014544  012724  014646                  MOV     #ISR67, (R4)+   ;SET UP TELEPRINTER INTERUPT VECTOR
3146  014550  012714  000340                  MOV     #340,   (R4)
3147  014554  032737  004000  000422          BIT     #SW11,@#$SWREG  ;TEST FOR ITERATIONS
3148  014562  001005                          BNE     1$              ;BRANCH TO AVOID HANG UP
3149  014564  000004  000473                  TYPE,   RETURN+1        ;RETURN+1 CAN BE REPLACED WITH THE ADDRESS OF RETURN
3150                                                                  ;TO TYPE CARRIAGE RETURN, LINE FEED
3151  014570  012767  014576  163672          MOV     #.+6,   LAD$    ;RESET LOOP ADDRESS
3152  014576  004567  001370          1$:     JSR     R5,     PUSHR   ;PUSH 4 WORDS ONTO R1 STACK, SET PRIORITY
3153  014602  000177  134543                  .WORD   000177,134543   ;SECOND OPERAND ON TOP
3154  014606  035700  143235                  .WORD   035700,143235   ;FIRST OPERAND ON BOTTOM
3155  014612  000143                          .WORD   143             ;PROCESSOR PRIORITY LEVEL
3156  014614  016442  000340                  .WORD   TRAPER, 340     ;FIS TRAP VECTOR
3157  014620  012701  000510                  MOV     #STACK0,R1      ;SET UP STACK POINTER
3158  014624  012767  000030  163702          MOV     #30,TEMP
3159  014632  112777  000100  163710          MOVB    #100,   @$TPS   ;SET TTY INTERUPT ENABLE
3160
3161  014640  075001                  RTA67:  FADD    R1              ;FLOATING ADD ON THE STACK
3162  014642  024141                          CMP     -(R1),  -(R1)   ;RESET THE STACK POINTER FOR NEXT PASS
3163  014644  000775                          BR      RTA67           ;REPEAT UNTIL INTERUPTED
3164
3165  014646  105077  163676          ISR67:  CLRB    @$TPS           ;CLEAR THE INTERUPT ENABLE
3166  014652  022716  014640                  CMP     #RTA67, (SP)    ;CHECK IF INTERUPT AT FIS INSTR.
3167  014656  001421                          BEQ     3$              ;BRANCH IF IT DID
3168  014660  022766  014640  000004          CMP     #RTA67, 4(SP)   ;CHECK FOR INTERUPT WITH T-BIT SET
3169  014666  001420                          BEQ     4$              ;BRANCH IF IT DID
3170  014670  112777  000015  163654  1$:     MOVB    #15,    @$TPB   ;CONTINUE TO TYPE "CR"
3171  014676  105777  163646          2$:     TSTB    @$TPS           ;LOOP HERE UNTILL DONE BIT COMES ON
3172  014702  100375                          BPL     2$
3173  014704  112777  000015  163640          MOVB    #15,    @$TPB   ;TYPE ANOTHER "CR"
3174  014712  012777  000100  163630          MOV     #100,   @$TPS   ;SET TTY INTERUPT ENABLE
3175  014720  000002                          RTI
3176
3177  014722  004767  001326          3$:     JSR     PC,     POPER   ;SAVE ALL THE STUFF ON THE STACK
3178  014726  000403                          BR      5$
3179
3180  014730  022626                  4$:     CMP     (SP)+,  (SP)+   ;RESET THE STACK TO IGNORE THE TRACE TRAP
3181  014732  004767  001322                  JSR     PC,     POPER1  ;POP ALL THE STUFF OFF THE STACK
3182  014736  005746                  5$:     TST     -(SP)           ;SAVE PSW FOR FUTURE RTI
3183  014740  012746  014640                  MOV     #RTA67,-(SP)    ;PLACE THE RTI ADDRESS BACK IN SP
3184  014744  022706  000574                  CMP     #BEGIN-4,SP     ;CHECK THE STACK POINTER
3185  014750  001407                          BEQ     6$              ;BRANCH IF OK
3186  014752  010667  163456                  MOV     SP,     $SP     ;SAVE FOR TYPING
```

```
3187  014756  012706  000574                  MOV     #BEGIN-4,SP     ;RESTORE THE STACK POINTER
3188  014762  104000                          HLT                     ;STACK POINTER FOULED UP
3189  014764  000525                          525                     ;THE ERROR NUMBER IS 525
3190  014766  000457                          BR      END67           ;SKIP REST OF TEST
3191
3192  014770  010167  163440          6$:     MOV     R1,     $SP     ;SAVE STACK POINTER
3193  014774  122767  000344  163430          CMPB    #344,   $PSW    ;CHECK PS AFTER INTERUPT
3194  015002  001402                          BEQ     .+6             ;BRANCH IF OK
3195  015004  104000                          HLT                     ;PS AFTER INTERUPT NOT EQUAL TO LVLA
3196  015006  000526                          526                     ;THE ERROR NUMBER IS 526
3197
3198  015010  022767  000510  163416          CMP     #STACK0,$SP     ;CHECK THE STACK POINTER (R1)
3199  015016  001402                          BEQ     .+6             ;BRANCH IF OK
3200  015020  104000                          HLT                     ;STACK POINTER (R1) NOT EQUAL TO #STACK0
3201  015022  000527                          527                     ;THE ERROR NUMBER IS 527
3202
3203  015024  022767  014640  163404          CMP     #RTA67, ANS1    ;CHECK FIS TRAP RETURN ADDRESS
3204  015032  001402                          BEQ     .+6             ;BRANCH IF OK
3205  015034  104001                          HLT+1                   ;FIS TRAP AT WRONG ADDRESS
3206  015036  000530                          530                     ;THE ERROR NUMBER IS 530
3207
3208
3209  015040  022767  000177  163374          CMP     #000177,ANS3    ;CHECK DATA FROM THE STACK
3210  015046  001402                          BEQ     .+6             ;BRANCH IF OK
3211  015050  104004                          HLT+4                   ;DATA ON STACK (000177) CHANGED
3212  015052  000531                          531                     ;THE ERROR NUMBER IS 531
3213
3214  015054  022767  134543  163362          CMP     #134543,ANS4    ;CHECK DATA FROM STACK
3215  015062  001402                          BEQ     .+6             ;BRANCH IF OK
3216  015064  104004                          HLT+4                   ;DATA ON STACK (134543) CHANGED
3217  015066  000532                          532                     ;THE ERROR NUMBER IS 532
3218
3219  015070  022767  035700  163350          CMP     #035700,ANS5    ;CHECK DATA FROM STACK
3220  015076  001402                          BEQ     .+6             ;BRANCH IF OK
3221  015100  104006                          HLT+6                   ;DATA ON STACK (035700) CHANGED
3222  015102  000533                          533                     ;THE ERROR NUMBER IS 533
3223
3224  015104  022767  143235  163336          CMP     #143235,ANS6    ;CHECK DATA FROM STACK
3225  015112  001402                          BEQ     .+6             ;BRANCH IF OK
3226  015114  104006                          HLT+6                   ;DATA ON STACK (143235) CHANGED
3227  015116  000534                          534                     ;THE ERROR NUMBER IS 534
3228
3229  015120  005367  163410                  DEC     TEMP            ;STAY IN THE LOOP FOR 30 TIMES
3230  015124  001261                          BNE     1$
3231
3232  015126  022626                  END67:  CMP     (SP)+,  (SP)+   ;RESTORE STACK POINTER TO 500
3233  015130  122767  000067  163246          CMPB    #67,    $TESTN  ;CHECK THE TEST NUMBER
3234  015136  001402                          BEQ     .+6             ;BRANCH IF OK
3235  015140  104000                          HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
3236  015142  000535                          535                     ;THE ERROR NUMBER IS 535
3237  015144                                  MTPS    #340
3238  015144  106427                          .WORD   106400!..C
3239
```

```
3240
3241                            ;*************************************************************************
3242                            ;TEST 70:       TEST THAT FIS ABORTS PROPERLY WHEN INTERUPTED
3243                            ;       107070,070707 * 040200,000000 = 107070,070707
3244                            ;       PS = .PS,       STACK POINTER = R0
3245                            ;*************************************************************************
3246
3247  015150  104400                        SCOPE
3248  015152  132737  000040  000421  TST70: BITB    #40,@#$ENVM
3249  015160  001173                        BNE     END70+2         ;EXIT THIS TEST IF BIT 5 OF $ENVM IS HIGH
3250  015162  013704  000546                MOV     @#TTYOUT,R4
3251  015166  012724  015270                MOV     #ISR70, (R4)+   ;SET UP TELEPRINTER INTERUPT VECTOR
3252  015172  012714  000340                MOV     #340,   (R4)
3253  015176  032737  004000  000422        BIT     #SW11,@#$SWREG  ;TEST FOR ITERATIONS
3254  015204  001005                        BNE     1$              ;BRANCH TO AVOID HANG UP
3255  015206  000004  000473                TYPE,   RETURN+1        ;RETURN+1 CAN BE REPLACED WITH THE ADDRESS OF RETURN
3256                                                                ;TO TYPE CARRIAGE RETURN, LINE FEED
3257  015212  012767  015220  163250        MOV     #.+6,   LAD$    ;RESET LOOP ADDRESS
3258  015220  004567  000746        1$:     JSR     R5,     PUSHR   ;PUSH 4 WORDS ONTO R0 STACK, SET PRIORITY
3259  015224  040200  000000                .WORD   040200,000000   ;SECOND OPERAND ON TOP
3260  015230  107070  070707                .WORD   107070,070707   ;FIRST OPERAND ON BOTTOM
3261  015234  000100                        .WORD   100             ;PROCESSOR PRIORITY LEVEL
3262  015236  016442  000340                .WORD   TRAPER, 340     ;FIS TRAP VECTOR
3263  015242  012700  000510                MOV     #STACK0,R0      ;SET UP STACK POINTER
3264  015246  012767  000030  163260        MOV     #30,TEMP
3265  015254  112777  000100  163266        MOVB    #100,   @$TPS   ;SET TTY INTERUPT ENABLE
3266
3267  015262  075020                RTA70:  FMUL    R0              ;FLOATING MULTIPLY ON THE STACK
3268  015264  024040                        CMP     -(R0),  -(R0)   ;RESET THE STACK POINTER FOR NEXT PASS
3269  015266  000775                        BR      RTA70           ;REPEAT UNTIL INTERUPTED
3270
3271  015270  105077  163254        ISR70:  CLRB    @$TPS           ;CLEAR THE INTERUPT ENABLE
3272  015274  022716  015262                CMP     #RTA70, (SP)    ;CHECK IF INTERUPT AT FIS INSTR.
3273  015300  001421                        BEQ     3$              ;BRANCH IF IT DID
3274  015302  022766  015262  000004        CMP     #RTA70, 4(SP)   ;CHECK FOR INTERUPT WITH T-BIT SET
3275  015310  001420                        BEQ     4$              ;BRANCH IF IT DID
3276  015312  112777  000015  163232  1$:   MOVB    #15,    @$TPB   ;CONTINUE TO TYPE "CR"
3277  015320  105777  163224        2$:     TSTB    @$TPS           ;LOOP HERE UNTILL DONE BIT COMES ON
3278  015324  100375                        BPL     2$
3279  015326  112777  000015  163216        MOVB    #15,    @$TPB   ;TYPE ANOTHER "CR"
3280  015334  012777  000100  163206        MOV     #100,   @$TPS   ;SET TTY INTERUPT ENABLE
3281  015342  000002                        RTI
3282
3283  015344  004767  000704        3$:     JSR     PC,     POPER   ;SAVE ALL THE STUFF ON THE STACK
3284  015350  000403                        BR      5$
3285
3286  015352  022626                4$:     CMP     (SP)+,  (SP)+   ;RESET THE STACK TO IGNORE THE TRACE TRAP
3287  015354  004767  000700                JSR     PC,     POPER1  ;POP ALL THE STUFF OFF THE STACK
3288  015360  005746                5$:     TST     -(SP)           ;SAVE PSW FOR FUTURE RTI
3289  015362  012746  015262                MOV     #RTA70,-(SP)    ;PLACE THE RTI ADDRESS BACK IN SP
3290  015366  022706  000574                CMP     #BEGIN-4,SP     ;CHECK THE STACK POINTER
3291  015372  001407                        BEQ     6$              ;BRANCH IF OK
3292  015374  010667  163034                MOV     SP,     $SP     ;SAVE FOR TYPING
3293  015400  012706  000574                MOV     #BEGIN-4,SP     ;RESTORE THE STACK POINTER
3294  015404  104000                        HLT                     ;STACK POINTER FOULED UP
3295  015406  000536                        536                     ;THE ERROR NUMBER IS 536
```

```
3296  015410  000456                          BR      END70           ;SKIP REST OF TEST
3297
3298  015412  010067  163016          6$:     MOV     R0,     $SP     ;SAVE STACK POINTER
3299  015416  122767  000344  163006          CMPB    #344,   $PSW    ;CHECK PS AFTER INTERUPT
3300  015424  001402                          BEQ     .+6             ;BRANCH IF OK
3301  015426  104000                          HLT                     ;PS AFTER INTERUPT NOT EQUAL TO LVLA
3302  015430  000537                          537                     ;THE ERROR NUMBER IS 537
3303
3304  015432  022767  000510  162774          CMP     #STACK0,$SP     ;CHECK THE STACK POINTER (R0)
3305  015440  001402                          BEQ     .+6             ;BRANCH IF OK
3306  015442  104000                          HLT                     ;STACK POINTER (R0) NOT EQUAL TO #STACK0
3307  015444  000540                          540                     ;THE ERROR NUMBER IS 540
3308
3309  015446  022767  015262  162762          CMP     #RTA70, ANS1    ;CHECK FIS TRAP RETURN ADDRESS
3310  015454  001402                          BEQ     .+6             ;BRANCH IF OK
3311  015456  104001                          HLT+1                   ;FIS TRAP AT WRONG ADDRESS
3312  015460  000541                          541                     ;THE ERROR NUMBER IS 541
3313
3314
3315  015462  022767  040200  162752          CMP     #040200,ANS3    ;CHECK DATA FROM THE STACK
3316  015470  001402                          BEQ     .+6             ;BRANCH IF OK
3317  015472  104004                          HLT+4                   ;DATA ON STACK (040200) CHANGED
3318  015474  000542                          542                     ;THE ERROR NUMBER IS 542
3319
3320  015476  005767  162742                  TST     ANS4            ;CHECK DATA FROM STACK
3321  015502  001402                          BEQ     .+6             ;BRANCH IF OK
3322  015504  104004                          HLT+4                   ;DATA ON STACK (000000) CHANGED
3323  015506  000543                          543                     ;THE ERROR NUMBER IS 543
3324
3325  015510  022767  107070  162730          CMP     #107070,ANS5    ;CHECK DATA FROM STACK
3326  015516  001402                          BEQ     .+6             ;BRANCH IF OK
3327  015520  104006                          HLT+6                   ;DATA ON STACK (107070) CHANGED
3328  015522  000544                          544                     ;THE ERROR NUMBER IS 544
3329
3330  015524  022767  070707  162716          CMP     #070707,ANS6    ;CHECK DATA FROM STACK
3331  015532  001402                          BEQ     .+6             ;BRANCH IF OK
3332  015534  104006                          HLT+6                   ;DATA ON STACK (070707) CHANGED
3333  015536  000545                          545                     ;THE ERROR NUMBER IS 545
3334
3335  015540  005367  162770                  DEC     TEMP            ;STAY IN THE LOOP FOR 30 TIMES
3336  015544  001262                          BNE     1$
3337
3338  015546  022626                  END70:  CMP     (SP)+,  (SP)+   ;RESTORE STACK POINTER TO 500
3339  015550  122767  000070  162626          CMPB    #70,    $TESTN  ;CHECK THE TEST NUMBER
3340  015556  001402                          BEQ     .+6             ;BRANCH IF OK
3341  015560  104000                          HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
3342  015562  000546                          546                     ;THE ERROR NUMBER IS 546
3343  015564                                  MTPS    #340
3344  015564  106427                          .WORD   106400!..C
3345
```

```
3346  015570  012767  000377  162740          MOV     #377,   TIMES   ;SET NUMBER OF ITERATIONS TO 377
3347  015576  010477  162744                  MOV     R4,     @TTYOUT ;RESTORE TTY INTERUPT VECTOR
3348  015602  005014                          CLR     (R4)
3349
```

```
3350                                    ;*********************************************************************
3351
3352                                    .SBTTL  END OF PASS ROUTINE
3353
3354                                    ;*INCREMENT THE PASS NUMBER ($PASS)
3355                                    ;*TYPE "END PASS"
3356                                    ;*IF THERES A MONITOR GO TO IT
3357                                    ;*IF THERE ISN'T JUMP TO BEGIN
3358                                    ;*IF IT IS DESIRED TO HAVE A BELL INDICATE THE "END OF PASS" LOCATION
3359                                    ;*SENDMG CAN BE CHANGED TO 7.
3360
3361  015604                            $EOP:
3362  015604  104400                           SCOPE
3363  015606  005267  162574                  INC     $PASS           ;;INCREMENT THE PASS NUMBER
3364  015612  042767  100000  162566          BIC     #100000,$PASS   ;;DON'T ALLOW A NEG. NUMBER
3365  015620  005327                          DEC     (PC)+           ;;LOOP?
3366  015622  000001                  $EOPCT: .WORD   1
3367  015624  003015                          BGT     $DOAGN          ;;YES
3368  015626  012737                          MOV     (PC)+,@(PC)+    ;;RESTORE COUNTER
3369  015630  000001                  $ENDCT: .WORD   1
3370  015632  015622                          $EOPCT
3371  015634  000004  015664                  TYPE    ,$ENDMG         ;;TYPE "END PASS"
3372  015640                          $GET42:
3373
3374  015640  013700  000042                  MOV     @#42,R0         ;;GET MONITOR ADDRESS
3375  015644  001405                          BEQ     $DOAGN          ;;BRANCH IF NO MONITOR
3376  015646  000005                          RESET                   ;;CLEAR THE WORLD
3377  015650  004710                  $ENDAD: JSR     PC,(R0)         ;;GO TO MONITOR
3378  015652  000240                          NOP                     ;;SAVE ROOM
3379  015654  000240                          NOP                     ;;FOR
3380  015656  000240                          NOP                     ;;ACT11
3381  015660                          $DOAGN:
3382  015660  000137  000600                  JMP     @#BEGIN         ;;RETURN
3383  015664  005015  047105  020104  $ENDMG: .ASCII  <15><12>/END PASS/
3384  015672  040520  051523
3385  015676     377     377     000  $ENULL: .BYTE   -1,-1,0         ;;NULL CHARACTER STRING
3386          015702                          .EVEN
3387
3388  015630  000001                  ENDCT:  1
```

```
3389                                    ;*****************************************************************
3390
3391                                    .SBTTL  SCOPE ROUTINE
3392
3393  015702  032737  000400  000422  SCOPE$: BIT     #SW08,@#$SWREG  ;KILL LDUB OR LOOP ON SPEC. TEST
3394  015710  001404                          BEQ     1$
3395  015712  123767  000422  162464          CMPB    @#$SWREG,$TESTN ;ON RIGHT TEST?   *SW7-0*
3396  015720  001431                          BEQ     OVER$
3397  015722  032737  040000  000422  1$:     BIT     #SW14,@#$SWREG  ;LOOP ON TEST
3398  015730  001023                          BNE     KIT$
3399  015732  032737  004000  000422          BIT     #SW11,@#$SWREG  ;KILL ITERATIONS
3400  015740  001412                          BEQ     SVLAD$
3401  015742  105767  162535                  TSTB    $ICNT
3402  015746  001404                          BEQ     2$              ;BRANCH IF FIRST
3403  015750  126767  162562  162525          CMPB    TIMES,$ICNT     ;DONE?
3404  015756  001010                          BNE     KIT$            ;BRANCH IF NOT
3405  015760  112767  000001  162515  2$:     MOVB    #1,$ICNT        ;FIRST ITERATION
3406  015766  105267  162412          SVLAD$: INCB    $TESTN          ;COUNT TEST NUMBERS
3407  015772  011667  162472                  MOV     (6),LAD$        ;SAVE LOOP ADDRESS
3408  015776  000002                          RTI                     ;RETURN
3409
3410  016000  105267  162477          KIT$:   INCB    $ICNT
3411  016004  005767  162460          OVER$:  TST     LAD$            ;FIRST ONE?
3412  016010  001766                          BEQ     SVLAD$
3413  016012  016716  162452                  MOV     LAD$,(6)        ;FUDGE RETURN ADDRESS
3414  016016  000002                          RTI                     ;FIXES PS
3415
```

```
3416
3417                                    ;SUBROUTINE TO PUSH 4 WORDS ONTO THE STACK
3418
3419  016020  005726                    PUSHS:  TST     (SP)+           ;POP STACK BY 1
3420  016022  062705  000010                    ADD     #10,    R5      ;POINT TO END OF DATA
3421  016026  014546                            MOV     -(R5),  -(SP)   ;PUSH DATA ONTO THE STACK
3422  016030  014546                            MOV     -(R5),  -(SP)   ;PUSH DATA ONTO THE STACK
3423  016032  014546                            MOV     -(R5),  -(SP)   ;PUSH DATA ONTO THE STACK
3424  016034  014546                            MOV     -(R5),  -(SP)   ;PUSH DATA ONTO THE STACK
3425  016036  062705  000010                    ADD     #10,    R5      ;POINT TO END OF DATA
3426  016042                                    MTPS    (R5)+           ;SET THE PROCESSOR STATUS
3427  016042  106425                            .WORD   106400!..C
3428  016044  005205                            INC     R5
3429  016046  012577  162412                    MOV     (R5)+,  @FISVEC ;SET UP FIS ERROR TRAP VECTOR
3430  016052  012577  162410                    MOV     (R5)+,  @FISLVL ;TRAP STATUS
3431  016056  000115                            JMP     (R5)            ;RETURN
3432
3433
3434                                    ;SUBROUTINE TO POP 2 WORDS OFF THE STACK
3435                                    ;ALSO SAVES THE PROCESSOR STATUS WORD (EXCEPT T BIT)
3436
3437  016060                            POPS:   MFPS    $PSW            ;SAVE PROCESSOR STATUS WORD
3438  016060  106767                            .WORD   106700!..C
3439  016064  042767  000020  162340            BIC     #20,    $PSW    ;CLEAR T-BIT
3440  016072  012604                            MOV     (SP)+,  R4      ;SAVE RTS ADDRESS
3441  016074  012667  162336                    MOV     (SP)+,  ANS1    ;SAVE THE ANSWER
3442  016100  012667  162334                    MOV     (SP)+,  ANS2    ;
3443  016104  010667  162324                    MOV     SP,     $SP     ;SAVE THE STACK POINTER
3444  016110  000114                            JMP     (R4)            ;RETURN
3445
3446
3447                                    ;SUBROUTINE TO POP 6 WORDS OFF THE STACK.
3448                                    ;THE FIRST TWO WERE PUT ON BY THE ERROR TRAP,
3449                                    ;THE LAST FOUR WERE THE ORIGINAL INPUT DATA.
3450                                    ;ALSO SAVES THE PS AND STACK POINTER.
3451
3452  016112                            POPES:  MFPS    $PSW            ;SAVE PROCESSOR STATUS WORD
3453  016112  106767                            .WORD   106700!..C
3454  016116  012604                            MOV     (SP)+,  R4      ;SAVE RTS ADDRESS
3455  016120  012667  162312                    MOV     (SP)+,  ANS1    ;SAVE RTI ADDRESS
3456  016124  011667  162310                    MOV     (SP),   ANS2    ;SAVE RTI STATUS
3457  016130  042767  000020  162302            BIC     #20,    ANS2    ;CLEAR THE T-BIT
3458  016136  012746  016144                    MOV     #1$,    -(SP)
3459  016142  000002                            RTI                     ;RESTORE THE PROCESSOR STATUS
3460  016144  012667  162272            1$:     MOV     (SP)+,  ANS3    ;SAVE DATA
3461  016150  012667  162270                    MOV     (SP)+,  ANS4    ;
3462  016154  012667  162266                    MOV     (SP)+,  ANS5    ;
3463  016160  012667  162264                    MOV     (SP)+,  ANS6    ;
3464  016164  010667  162244                    MOV     SP,     $SP     ;SAVE SP
3465  016170  000114                            JMP     (R4)            ;RTS
3466
3467                                    ;SUBROUTINE TO PUSH 4 WORDS ONTO THE STACK
3468
3469  016172  012704  000510            PUSHR:  MOV     #STACK0,R4      ;SET R4 TO STACK
3470  016176  012524                            MOV     (R5)+,  (R4)+   ;PUT DATA ON STACK
3471  016200  012524                            MOV     (R5)+,  (R4)+   ;
```

```
3472  016202  012524                          MOV     (R5)+,  (R4)+   ;
3473  016204  012524                          MOV     (R5)+,  (R4)+   ;
3474  016206                                  MTPS    (R5)+           ;SET THE PROCESSOR STATUS
3475  016206  106425                          .WORD   106400!..C
3476  016210  005205                          INC     R5
3477  016212  012577  162246                 MOV     (R5)+,  @FISVEC ;SET UP FIS ERROR TRAP VECTOR
3478  016216  012577  162244                 MOV     (R5)+,  @FISLVL ;TRAP STATUS
3479  016222  000205                          RTS     R5              ;RETURN
3480
3481
3482                                  ;SUBROUTINE TO POP 2 WORDS OFF THE STACK
3483                                  ;ALSO SAVES THE PROCESSOR STATUS WORD (EXCEPT T BIT)
3484
3485  016224                          POPR:   MFPS    $PSW            ;SAVE PROCESSOR STATUS WORD
3486  016224  106767                          .WORD   106700!..C
3487  016230  042767  000020  162174          BIC     #20,    $PSW    ;CLEAR T-BIT
3488  016236  016767  162252  162172          MOV     STACK4, ANS1    ;SAVE THE ANSWER
3489  016244  016767  162246  162166          MOV     STACK6, ANS2    ;
3490  016252  000207                          RTS     PC
3491
3492
3493                                  ;SUBROUTNE TO POP 6 WORDS OFF THE STACKS.
3494                                  ;THE TWO OFF THE R6 STACK WERE PUT ON BY THE ERROR TRAP.
3495                                  ;THE FOUR OFF THE SOFTWARE STACK WERE THE ORIGINAL INPUT DATA.
3496                                  ;ALSO SAVES THE PS AND STACK POINTER AFTER THE FIS TRAP.
3497
3498  016254                          POPER:  MFPS    $PSW            ;SAVE PROCESSOR STATUS WORD
3499  016254  106767                          .WORD   106700!..C
3500  016260  012667  000056          POPER1: MOV     (SP)+,  SAVRTS  ;SAVE RTS ADDRESS
3501  016264  012667  162146                  MOV     (SP)+,  ANS1    ;SAVE RTI ADDRESS
3502  016270  011667  162144                  MOV     (SP),   ANS2    ;SAVE RTI STATUS
3503  016274  042767  000020  162136          BIC     #20,    ANS2    ;CLEAR THE T-BIT
3504  016302  012746  016310                  MOV     #1$,    -(SP)
3505  016306  000002                          RTI                     ;RESTORE PROCESSOR STATUS
3506  016310  016767  162174  162124  1$:     MOV     STACK0, ANS3    ;SAVE DATA
3507  016316  016767  162170  162120          MOV     STACK2, ANS4    ;
3508  016324  016767  162164  162114          MOV     STACK4, ANS5    ;
3509  016332  016767  162160  162110          MOV     STACK6, ANS6    ;
3510  016340  000137                          JMP     @(PC)+          ;SIMULATED RTS
3511  016342  000000                  SAVRTS: 0
3512
3513                                  ;SUBROUTINE TO PUSH 4 WORDS ONTO THE PC STACK
3514
3515  016344  012504                  PUSH7:  MOV     (R5)+,  R4      ;SET R4 TO STACK
3516  016346  012524                          MOV     (R5)+,  (R4)+   ;PUT DATA ON STACK
3517  016350  012524                          MOV     (R5)+,  (R4)+   ;
3518  016352  012524                          MOV     (R5)+,  (R4)+   ;
3519  016354  012524                          MOV     (R5)+,  (R4)+   ;
3520  016356                                  MTPS    (R5)+           ;SET THE PROCESSOR STATUS
3521  016356  106425                          .WORD   106400!..C
3522  016360  005205                          INC     R5
3523  016362  012577  162076                 MOV     (R5)+,  @FISVEC ;SET UP FIS ERROR TRAP VECTOR
3524  016366  012577  162074                 MOV     (R5)+,  @FISLVL ;TRAP STATUS
3525  016372  000205                          RTS     R5              ;RETURN
3526
3527                                  ;SUBROUTINE TO POP 4 WORDS OFF THE PC "STACK"
```

```
3528                                        ;ALSO SAVES THE PROCESSOR STATUS WORD (EXCEPT T BIT)
3529
3530  016374                                POP7:   MFPS    $PSW            ;SAVE PROCESSOR STATUS WORD
3531  016374  106767                                .WORD   106700!..C
3532  016400  042767  000020  162024              BIC     #20,    $PSW    ;CLEAR T-BIT
3533  016406  011600                              MOV     (SP),   R0      ;GET RETURN ADDRESS
3534  016410  162700  000014                      SUB     #14,    R0      ;POINT TO TOP OF "PC STACK"
3535  016414  012067  162016                      MOV     (R0)+,  ANS1    ;SAVE 1ST HALF INPUT DATA
3536  016420  012067  162014                      MOV     (R0)+,  ANS2    ;SAVE 2ND HALF INPUT DATA
3537  016424  010067  162004                      MOV     R0,     $SP     ;SAVE ASSUMED END PC "STACK POINTER"
3538  016430  012067  162006                      MOV     (R0)+,  ANS3    ;SAVE 1ST HALF OF ANSWER
3539  016434  012067  162004                      MOV     (R0)+,  ANS4    ;SAVE 2ND HALF OF ANSWER
3540  016440  000207                              RTS     PC
3541
3542                                        ;ERRONIOUS TRAP SERVICE ROUTINE
3543
3544  016442  104000                        TRAPER: HLT                     ;FIS SHOULDN'T HAVE TRAPED
3545  016444  000547                              547                     ;THE ERROR NUMBER IS 547
3546  016446  000002                              RTI
3547
```

```
3548                                    ;*****************************************************************
3549                                    .SBTTL  HLT ROUTINE (ERROR TYPEOUT)
3550  016450  032737  002000  000422    HLT$:   BIT     #SW10,@#$SWREG  ;SHOULD IT RING THE BELL ON ERROR?
3551  016456  001402                            BEQ     1$              ;NO - SKIP
3552  016460  000004  000504                    TYPE    ,$BELL          ;RING BELL
3553  016464  005267  161772            1$:     INC     ERRORS          ;COUNT THE NUMBER OF ERRORS
3554  016470  032737  020000  000422            BIT     #SW13,@#$SWREG  ;SKIP TYPEOUT IF SET
3555  016476  001023                            BNE     2$              ;SKIP TYPEOUTS
3556  016500  000004  000472                    TYPE    ,RETURN
3557  016504  013637  000402                    MOV     @(6)+,@#$FATAL  ;PLACE THE ERROR NUMBER IN LOCATION $FATAL
3558  016510  014667  161714                    MOV     -(6),HLTAD$     ;PUT ADDRESS OF INSTRUCTION ON STACK
3559  016514  162767  000002  161706            SUB     #2,HLTAD$
3560  016522  017605  000000                    MOV     @(6),TTY        ;TYPE @(6) IN OCTAL
3561  016526  004767  000124                    JSR     %7,PRINTR       ;TYPE LEADING ZERO'S
3562  016532  062716  000002                    ADD     #2,(6)          ;ADJUST THE RETURN ADDRESS
3563  016536  000004  000500                    TYPE    ,SPACE+3
3564  016542  004767  000046                    JSR     PC,ERROR$       ;GO TO USER ERROR ROUTINE
3565  016546  105767  161646            2$:     TSTB    $ENV            ;ARE WE RUNNING UNDER APT?
3566  016552  001403                            BEQ     4$              ;IF NOT THEN GO TO 4$
3567  016554  005237  000400                    INC     @#$MSGTY        ;OTHERWISE INFORM APT
3568  016560  000777                            BR      .               ;AND LOOP
3569  016562  005737  000422            4$:     TST     @#$SWREG        ;HALT ON ERROR
3570  016566  100001                            BPL     .+4             ;SKIP IF CONTINUE
3571  016570  000000                            HALT                    ;HALT ON ERROR!
3572  016572  032737  001000  000422            BIT     #SW09,@#$SWREG  ;CHECK FOR INHIBIT LOOP ON ERROR
3573  016600  001001                            BNE     .+4             ;SKIP IF LOOP ON ERROR
3574  016602  000002                            RTI
3575  016604  105067  161673                    CLRB    $ICNT
3576  016610  000167  177164                    JMP     KIT$            ;LOOP ON TEST UNTIL NO ERRORS
3577
3578
```

```
3579
3580                                    ;*****************************************************************
3581
3582                                    .SBTTL  USER ERROR ROUTINE
3583
3584  016614  117767  161610  161716  ERROR$: MOVB    @HLTAD$,TYPCNT  ;TYPE COUNT IS LOW BYTE OF HLT
3585  016622  062767  000002  161710          ADD     #2,     TYPCNT  ;TYPE COUNT = X+2
3586  016630  012703  000430                  MOV     #HLTAD$,R3      ;TOP OF DATA TO BE TYPED
3587  016634                          ERR1$:
3588  016634  012305                          MOV     (R3)+,TTY       ;TYPE (R3)+ IN OCTAL
3589  016636  004767  000014                  JSR     %7,PRINTR       ;TYPE LEADING ZERO'S
3590  016642  000004  000501                  TYPE,   SPACE+4         ;SPACE
3591  016646  105367  161666                  DECB    TYPCNT          ;CHECK FOR DONE
3592  016652  100370                          BPL     ERR1$           ;BRANCH IF NOT DONE
3593  016654  000207                          RTS     PC
3594
```

```
 3595  016656  112767  000001  161660  PRINTR: MOVB    #1,.PR          ;SET ZERO FILL SWITCH
 3596  016664  000402                  BR      .+6                     ;SKIP
 3597  016666  005067  161652          PRINTS: CLR     .PR             ;SUPRESS LEADING ZERO'S
 3598  016672  112767  177772  161645          MOVB    #-6,.PR+1       ;SET COUNT
 3599  016700  010446                          MOV     R4,-(6)         ;SAVE R4
 3600  016702  012704  017004                  MOV     #.PRBUF,R4      ;SET POINTER TO FIRST ASCII CHAR.
 3601  016706  105014                          CLRB    (4)             ;CLEAR FIRST BYTE
 3602  016710  000405                  BR      .PRF                    ;ROTATE FIRST BIT
 3603  016712  105014          .PRL:   CLRB    (4)             ;CLEAR BYTE OF CHARACTER
 3604  016714  006105                  ROL     TTY             ;ROTATE BIT INTO C
 3605  016716  106114                  ROLB    (4)             ;PACK IT
 3606  016720  006105                  ROL     TTY             ;ROTATE BIT INTO C
 3607  016722  106114                  ROLB    (4)             ;PACK IT
 3608  016724  006105          .PRF:   ROL     TTY             ;ROTATE BIT INTO C
 3609  016726  106114                  ROLB    (4)             ;PACK IT
 3610  016730  105714                  TSTB    (4)             ;IS IT ZERO?
 3611  016732  001402                  BEQ     .+6             ;SKIP INC
 3612  016734  105267  161604          INCB    .PR             ;SET FILL SWITCH
 3613  016740  105767  161600          TSTB    .PR             ;CHECK FILL SWITCH
 3614  016744  001402                  BEQ     .+6             ;SKIP BITSET
 3615  016746  152724  000060          BISB    #'0,(4)+        ;MAKE INTO ASCII CHAR
 3616  016752  105267  161567          INCB    .PR+1           ;INC COUNT
 3617  016756  001355                  BNE     .PRL            ;REPEAT
 3618  016760  022704  017004          CMP     #.PRBUF,R4      ;EMPTY BUFFER?
 3619  016764  001002                  BNE     .+6             ;SKIP IF NOT
 3620  016766  112724  000060          MOVB    #'0,(4)+        ;LOAD 1 ZERO
 3621  016772  105014                  CLRB    (4)             ;NULL TERMINATOR
 3622  016774  000004  017004          TYPE    .PRBUF          ;TYPE IT
 3623  017000  012604                  MOV     (6)+,R4         ;RESTORE R4
 3624  017002  000207                  RTS     PC              ;RETURN
 3625
 3626  017004  000004          .PRBUF: .BLKW   4               ;OUTPUT BUFFER
```

```
                                ;*****************************************************************

                                .SBTTL  POWER DOWN AND UP ROUTINES

                                ;POWER DOWN ROUTINE
017014  012737  017136  000024  $PWRDN: MOV     #$ILLUP,@#PWRVEC ;;SET FOR FAST UP
017022  012737  000340  000026          MOV     #340,@#PWRVEC+2 ;;PRIO:7
017030  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
017032  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
017034  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
017036  010346                          MOV     R3,-(SP)        ;;PUSH R3 ON STACK
017040  010446                          MOV     R4,-(SP)        ;;PUSH R4 ON STACK
017042  010546                          MOV     R5,-(SP)        ;;PUSH R5 ON STACK
017044  010667  000072                  MOV     SP,$SAVR6       ;;SAVE SP
017050  012737  017062  000024          MOV     #$PWRUP,@#PWRVEC ;;SET UP VECTOR
017056  000000                          HALT
017060  000776                          BR      .-2             ;;HANG UP

                                ;POWER UP ROUTINE
017062  016706  000054          $PWRUP: MOV     $SAVR6,SP       ;;GET SP
017066  005067  000050                  CLR     $SAVR6          ;;WAIT LOOP FOR THE TTY
017072  005267  000044          1$:     INC     $SAVR6          ;;WAIT FOR THE INC
017076  001375                          BNE     1$              ;;OF  WORD
017100  012605                          MOV     (SP)+,R5        ;;POP STACK INTO R5
017102  012604                          MOV     (SP)+,R4        ;;POP STACK INTO R4
017104  012603                          MOV     (SP)+,R3        ;;POP STACK INTO R3
017106  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
017110  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
017112  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
017114  012737  017014  000024          MOV     #$PWRDN,@#PWRVEC ;;SET UP THE POWER DOWN VECTOR
017122  012737  000340  000026          MOV     #340,@#PWRVEC+2 ;;PRIO:7
017130  000004                          TYPE                    ;REPORT THE POWER FAILURE
017132  017144                  $PWRMG: .WORD   $POWER          ;;POWER FAIL MESSAGE POINTER
017134  000002                          RTI
017136  000000                  $ILLUP: HALT                    ;;THE POWER UP SEQUENCE WAS STARTED
017140  000776                          BR      .-2             ;;  BEFORE THE POWER DOWN WAS COMPLETE
017142  000000                  $SAVR6: 0                       ;;PUT THE SP HERE
017144  005015  047520  042527  $POWER: .ASCIZ  <15><12>"POWER"
017152  000122
                                        .EVEN
```

```
3667
3668                                    ;*      TYPE OUT ROUTINE
3669                                    ;*      ----------------
3670                                    ;*
3671                                    ;*
3672                                    ;*      THIS ROUTINE IS USED TO TYPE ASCIZ MESSAGES
3673                                    ;*
3674
3675  017154  132737  000040  000421  $TYPE:  BITB    #40,@#$ENVM     ;HAS THE CONSOLE OUTPUTS BEEN SUPPRESSED?
3676  017162  001007                    BNE     3$              ;IF SO THEN RETURN FROM THE SUBROUTINE VIA 3$
3677  017164  010046                    MOV     R0,-(SP)        ;OTHERWISE SAVE R0
3678  017166  017600  000002            MOV     @2(SP),R0       ;GET THE ADDRESS OF THE ASSCIZ STRING
3679  017172  112046            2$:     MOVB    (R0)+,-(SP)     ;PUSH THE CHARACTER TO BE TYPED ONTO STACK
3680  017174  001005                    BNE     4$              ;BRANCH IF IT IS NOT THE TERMINATOR
3681  017176  005726                    TST     (SP)+
3682  017200  012600                    MOV     (SP)+,R0        ;OTHERWISE RESTORE THE STACK AND R0
3683  017202  062716  000002    3$:     ADD     #2,(SP)         ;ADJUST THE RETURN PC
3684  017206  000002                    RTI                     ;AND RETURN
3685
3686  017210  105777  161334    4$:     TSTB    @$TPS           ;IS THE PRINTER AVAILABLE?
3687  017214  100375                    BPL     4$              ;IF NOT THEN LOOP HERE
3688  017216  112677  161330            MOVB    (SP)+,@$TPB     ;OUT PUT THE CHARACTER
3689  017222  000763                    BR      2$              ;AND GO BACK
3690          000001            .END
```

```
A      = 015702           3388#   3389
ABASE  = 000000            100
ACDW1  = 000000            100
ACDW2  = 000000            100
ACPUOP= 000000            100     118
ADDW0  = 000000            100
ADDW1  = 000000            100
ADDW10= 000000            100
ADDW11= 000000            100
ADDW12= 000000            100
ADDW13= 000000            100
ADDW14= 000000            100
ADDW15= 000000            100
ADDW2  = 000000            100
ADDW3  = 000000            100
ADDW4  = 000000            100
ADDW5  = 000000            100
ADDW6  = 000000            100
ADDW7  = 000000            100
ADDW8  = 000000            100
ADDW9  = 000000            100
ADEVCT= 000000            100     109
ADEVM  = 000000            100
AENV   = 000000            100     114
AENVM  = 000000            100     115
AFATAL= 000000            100     106
AMADR1= 000000            100
AMADR2= 000000            100
AMADR3= 000000            100
AMADR4= 000000            100
AMAMS1= 000000            100
AMAMS2= 000000            100
AMAMS3= 000000            100
AMAMS4= 000000            100
AMSGAD= 000000            100     111
AMSGLG= 000000            100     112
AMSGTY= 000000            100     105
AMTYP1= 000000            100
AMTYP2= 000000            100
AMTYP3= 000000            100
AMTYP4= 000000            100
ANS1       000436          156#    157     263     309     355     401     447     493     539     585     631     677     723
                           769     815     859     921     993    1065    1137    1203    1249    1300    1346    1397    1443
                          1489    1535    1581    1627    1673    1724    1776    1848    1914    1960    2006    2052    2098
                          2144    2188    2242    2304    2376    2442    2488    2534    2580    2624    2686    2758    2830
                          2902    2976*   2990    3044    3117    3203    3309    3441*   3455*   3488*   3501*   3535*
ANS2       000440          158#    159     268     314     360     406     452     498     544     590     636     682     728
                           774     820     864     926     998    1070    1142    1208    1254    1305    1351    1402    1448
                          1494    1540    1586    1632    1678    1729    1781    1853    1919    1965    2011    2057    2103
                          2149    2193    2247    2309    2381    2447    2493    2539    2585    2629    2691    2763    2835
                          2907    2977*   2995    3049    3122    3442*   3456*   3457*   3489*   3502*   3503*   3536*
ANS3       000442          160#    161     869     931    1003    1075    1147    1786    1858    2198    2252    2314    2386
                          2634    2696    2768    2840    2917    3054    3209    3315    3460*   3506*   3538*
ANS4       000444          162#    874     936    1008    1080    1152    1791    1863    2203    2257    2319    2391    2639
                          2701    2773    2845    2922    3059    3214    3320    3461*   3507*   3539*
ANS5       000446          163#    941    1013    1085    1157    1796    1868    2324    2396    2706    2778    2850    2927
```

```
                    3064    3219    3325    3462*   3508*
ANS6    000450      164#     946    1018    1090    1162    1801    1873    2329    2401    2711    2783    2855    2932
                    3069    3224    3330    3463*   3509*
APASS = 000000       100     108
APRIOR= 000000       100
ASWREG= 000000       100     116
ATESTN= 000000       100     107
AUNIT = 000000       100     110
AUSWR = 000000       100     117
AVECT1= 000000       100
AVECT2= 000000       100
BEGIN   000600       206     212#    435     437     481     483     711     713     757     759     977     981     983
                    1121    1125    1127    1283    1285    1380    1382    1707    1709    1760    1764    1766    2086
                    2088    2360    2364    2366    2476    2478    2886    2890    2892    3184    3187    3290    3293
                    3382
ENDCT   015630      3388#
END1    001040       273#
END10   002254       595#
END11   002412       641#
END12   002550       687#
END13   002702       716     733#
END14   003036       762     779#
END15   003176       825#
END16   003350       879#
END17   003602       907     951#
END2    001200       319#
END20   004032       978     986    1023#
END21   004270      1051    1095#
END22   004522      1122    1130    1167#
END23   004662      1213#
END24   005016      1259#
END25   005152      1293    1310#
END26   005310      1356#
END27   005444      1390    1407#
END3    001334       365#
END30   005604      1453#
END31   005744      1499#
END32   006104      1545#
END33   006242      1591#
END34   006376      1637#
END35   006534      1683#
END36   006670      1717    1734#
END37   007122      1761    1769    1806#
END4    001470       411#
END40   007360      1834    1878#
END41   007514      1924#
END42   007654      1970#
END43   010010      2016#
END44   010144      2062#
END45   010300      2091    2108#
END46   010436      2154#
END47   010612      2208#
END5    001622       440     457#
END50   010766      2262#
END51   011222      2290    2334#
END52   011454      2361    2369    2406#
```

```
END53   011612            2452#
END54   011746            2481    2498#
END55   012104            2544#
END56   012240            2590#
END57   012414            2644#
END6    001754             486     503#
END60   012652            2672    2716#
END61   013110            2744    2788#
END62   013346            2816    2860#
END63   013600            2887    2895    2937#
END64   014016            3000#
END65   014274            3030    3073#
END66   014472            3103    3127#
END67   015126            3143    3190    3232#
END7    002114             549#
END70   015546            3249    3296    3338#
ERRORS  000462             167#   3553*
ERRORS  016614            3564    3584#
ERR1$   016634            3587#   3592
FISLVL  000466             169#   3430*   3478*   3524*
FISVEC  000464             168#   3429*   3477*   3523*
HLT   = 104000              56#     255     260     265     270     275     301     306     311     316     321     347     352
                           357     362     367     393     398     403     408     413     438     444     449     454     459
                           484     490     495     500     505     531     536     541     546     551     577     582     587
                           592     597     623     628     633     638     643     669     674     679     684     689     714
                           720     725     730     735     760     766     771     776     781     807     812     817     822
                           827     856     861     866     871     876     881     905     913     918     923     928     933
                           938     943     948     953     975     984     990     995    1000    1005    1010    1015    1020
                          1025    1049    1057    1062    1067    1072    1077    1082    1087    1092    1097    1119    1128
                          1134    1139    1144    1149    1154    1159    1164    1169    1195    1200    1205    1210    1215
                          1241    1246    1251    1256    1261    1291    1297    1302    1307    1312    1338    1343    1348
                          1353    1358    1388    1394    1399    1404    1409    1435    1440    1445    1450    1455    1481
                          1486    1491    1496    1501    1527    1532    1537    1542    1547    1573    1578    1583    1588
                          1593    1619    1624    1629    1634    1639    1665    1670    1675    1680    1685    1715    1721
                          1726    1731    1736    1758    1767    1773    1778    1783    1788    1793    1798    1803    1808
                          1832    1840    1845    1850    1855    1860    1865    1870    1875    1880    1906    1911    1916
                          1921    1926    1952    1957    1962    1967    1972    1998    2003    2008    2013    2018    2044
                          2049    2054    2059    2064    2089    2095    2100    2105    2110    2136    2141    2146    2151
                          2156    2185    2190    2195    2200    2205    2210    2239    2244    2249    2254    2259    2264
                          2288    2296    2301    2306    2311    2316    2321    2326    2331    2336    2358    2367    2373
                          2378    2383    2388    2393    2398    2403    2408    2434    2439    2444    2449    2454    2479
                          2485    2490    2495    2500    2526    2531    2536    2541    2546    2572    2577    2582    2587
                          2592    2621    2626    2631    2636    2641    2646    2670    2678    2683    2688    2693    2698
                          2703    2708    2713    2718    2742    2750    2755    2760    2765    2770    2775    2780    2785
                          2790    2814    2822    2827    2832    2837    2842    2847    2852    2857    2862    2884    2893
                          2899    2904    2914    2919    2924    2929    2934    2939    2981    2986    2992    2997    3002
                          3028    3036    3041    3046    3051    3056    3061    3066    3071    3075    3101    3109    3114
                          3119    3124    3129    3188    3195    3200    3205    3211    3216    3221    3226    3235    3294
                          3301    3306    3311    3317    3322    3327    3332    3341    3544
HLTADS  000430             150#     151    3558*   3559*   3584    3586
HLTS    016450             218     3550#
ISA20   003710             982      988#
ISA22   004376            1126     1132#
ISA37   006776            1765     1771#
ISA52   011330            2365     2371#
ISA63   013454            2891     2897#
```

```
ISR17   003436            897     909#
ISR20   003664            969     980#
ISR21   004120           1041    1053#
ISR22   004352           1113    1124#
ISR37   006752           1752    1763#
ISR40   007210           1824    1836#
ISR51   011054           2280    2292#
ISR52   011304           2352    2363#
ISR60   012502           2662    2674#
ISR61   012740           2734    2746#
ISR62   013176           2806    2818#
ISR63   013430           2878    2889#
ISR65   014124           3012    3032#
ISR66   014402           3084    3105#
ISR67   014646           3145    3165#
ISR70   015270           3251    3271#
KIT$    016000           3398    3404   3410#   3576
LAD$    000470            170#    230#  3151#  3257#  3407#  3411   3413
N     = 000071             77#    233    279#   325#   371#   417#   463#   509#   555#   601#   647#   693#   739#
                          785#    831#   885#   957#  1029#  1101#  1173#  1219#  1265#  1316#  1362#  1413#  1459#
                         1505#   1551#  1597#  1643#  1689#  1740#  1812#  1884#  1930#  1976#  2022#  2068#  2114#
                         2160#   2214#  2268#  2340#  2412#  2458#  2504#  2550#  2596#  2650#  2722#  2794#  2866#
                         2943#   2969   3005#  3077#  3131#  3134   3240#  3346#
OVER$   016004           3396    3411#
POPER   016254            909    1053   1836   2292   2674   2746   2818   3032   3105   3177   3283   3498#
POPER1  016260           3181    3287   3500#
POPES   016112            980    1124   1763   2363   2889   3452#
POPR    016224            251     297    343    389    527    573    619    665    803    903   1047   1191   1237
                         1334    1431   1477   1523   1569   1615   1661   1830   1902   1948   1994   2040   2132
                         2286    2430   2522   2568   2668   2740   2812   3026   3485#
POPS    016060            434     480    710    756    974   1118   1282   1379   1706   1757   2085   2357   2475
                         2883    3437#
POP7    016374            853    2182   2236   2618   3530#
PRINTR  016656           3561    3589   3595#
PRINTS  016666           3597#
PUSHR   016172            241     287    333    379    517    563    609    655    793    893   1037   1181   1227
                         1324    1421   1467   1513   1559   1605   1651   1820   1892   1938   1984   2030   2122
                         2276    2420   2512   2558   2658   2730   2802   3014   3086   3152   3258   3469#
PUSHS   016020            425     471    701    747    965   1109   1273   1370   1697   1748   2076   2348   2466
                         2874    3419#
PUSH7   016344            839    2168   2222   2604   3515#
PWRVEC= 000024             66#   3632*  3633*  3641*  3656*  3657*
RESTRT  000222            206#
RETURN  000472            171#   3149   3255   3556
RTA17   003420            903#    921
RTA20   003646            974#    993
RTA21   004102           1047#   1065
RTA22   004334           1118#   1137
RTA37   006734           1757#   1776
RTA40   007172           1830#   1848
RTA51   011036           2286#   2304
RTA52   011266           2357#   2376
RTA60   012464           2668#   2686
RTA61   012722           2740#   2758
RTA62   013160           2812#   2830
RTA63   013412           2883#   2902
```

```
RTA65   014106          3026#  3044
RTA66   014364          3098#  3117
RTA67   014640          3161#  3163  3166  3168  3183  3203
RTA70   015262          3267#  3269  3272  3274  3289  3309
SAVRTS  016342          3500*  3511#
SAVTPS  000506           177#
SCOPE = 104400            67#   240   286   332   378   424   470   516   562   608   654   700   746
                          792   838   892   964  1036  1108  1180  1226  1272  1323  1369  1420  1466
                         1512  1558  1604  1650  1696  1747  1819  1891  1937  1983  2029  2075  2121
                         2167  2221  2275  2347  2419  2465  2511  2557  2603  2657  2729  2801  2873
                         2952  3011  3083  3141  3247  3362
SCOPES  015702           220  3393#
SPACE   000475           172#  3563  3590
STACK0  000510           178#   185   246   292   338   384   522   568   614   660   798   898   916
                         1042  1060  1186  1232  1329  1426  1472  1518  1564  1610  1656  1825  1843
                         1897  1943  1989  2035  2127  2281  2299  2425  2517  2563  2663  2681  2735
                         2753  2807  2825  3157  3198  3263  3304  3469  3506
STACK1= 000511           185#
STACK2  000512           179#  3507
STACK4  000514           180#   258   304   350   396   534   580   626   672   810  1198  1244  1341
                         1438  1484  1530  1576  1622  1668  1909  1955  2001  2047  2139  2437  2529
                         2575  3488  3508
STACK6  000516           181#  3489  3509
STACK8  000520           182#
STAK10  000532           184#  2953  2984
STK16   003244           840    848#
STK47   010504          2169   2177#
STK50   010660          2223   2231#
STK57   012306          2605   2613#
SVLADS  015766          3400   3406#  3412
SW08  = 000400            35     75#  3393
SW09  = 001000            36     74#  3572
SW10  = 002000            37     73#  3550
SW11  = 004000            38     72#  3147  3253  3399
SW12  = 010000            39     71#
SW13  = 020000            40     70#  3554
SW14  = 040000            41     69#  3397
SW15  = 100000            42     68#
TEMP    000534           186#  3158*  3229*  3264*  3335*
TIMES   000536           187#   200*  3133*  3346*  3403
TRAPER  016442           245    291    337    383    429   475   521   567   613   659   705   751   797
                         844   1185  1231  1277  1328  1374  1425  1471  1517  1563  1609  1655  1701
                        1896  1942  1988  2034  2080  2126  2173  2227  2424  2470  2516  2562  2609
                        3018  3090  3156  3262  3544#
TSA13   002640           712    718#
TSA14   002772           758    764#
TSA25   005106          1284   1295#
TSA27   005400          1381   1392#
TSA36   006624          1708   1719#
TSA45   010234          2087   2093#
TSA5    001560           436    442#
TSA54   011702          2477   2483#
TSA6    001712           482    488#
TST1    000722           241#
TST10   002132           563#
TST11   002272           609#
```

```
TST12   002430            655#
TST13   002566            701#
TST14   002720            747#
TST15   003054            793#
TST16   003214            839#
TST17   003366            893#
TST2    001056            287#
TST20   003620            965#
TST21   004050           1037#
TST22   004306           1109#
TST23   004540           1181#
TST24   004700           1227#
TST25   005034           1273#
TST26   005170           1324#
TST27   005326           1370#
TST3    001216            333#
TST30   005462           1421#
TST31   005622           1467#
TST32   005762           1513#
TST33   006122           1559#
TST34   006260           1605#
TST35   006414           1651#
TST36   006552           1697#
TST37   006706           1748#
TST4    001352            379#
TST40   007140           1820#
TST41   007376           1892#
TST42   007532           1938#
TST43   007672           1984#
TST44   010026           2030#
TST45   010162           2076#
TST46   010316           2122#
TST47   010454           2168#
TST5    001506            425#
TST50   010630           2222#
TST51   011004           2276#
TST52   011240           2348#
TST53   011472           2420#
TST54   011630           2466#
TST55   011764           2512#
TST56   012122           2558#
TST57   012256           2604#
TST6    001640            471#
TST60   012432           2658#
TST61   012670           2730#
TST62   013126           2802#
TST63   013364           2874#
TST64   013616           2953#
TST65   014034           3012#
TST66   014312           3084#
TST67   014530           3142#
TST7    001772            517#
TST70   015152           3248#
TTYOUT  000546            191#   3144    3250    3347*
TYPCNT  000540            188#   3584*   3585*   3591*
TYPE  = 000004             76#   3149    3255    3371    3552    3556    3563    3590    3622    3658
```

```
YESRT   000542            189#   213
$APTHD  000430            136    142#   149
$BELL   000504            176#  3552
$CPUOP  000426            118#
$DEVCT  000410            109#   201
$DOAGN  015660           3367   3375   3381#
$ENDAD  015650             89   3377#
$ENDCT  015630           3369#  3388
$ENDMG  015664           3371   3383#
$ENULL  015676           3385#
$ENV    000420            114#   223   3565
$ENVM   000421            115#  3142   3248   3675
$EOP    015604           3361#
$EOPCT  015622           3366#  3370
$ETABL  000420            113#
$ETEND  000430            125#   148
$F    = 000550             78#   256   257#   261   262#   266   267#   271   272#   276   277#   302   303#
                          307    308#   312   313#   317   318#   322   323#   348   349#   353   354#   358
                          359#   363   364#   368   369#   394   395#   399   400#   404   405#   409   410#
                          414    415#   439   440#   445   446#   450   451#   455   456#   460   461#   485
                          486#   491   492#   496   497#   501   502#   506   507#   532   533#   537   538#
                          542    543#   547   548#   552   553#   578   579#   583   584#   588   589#   593
                          594#   598   599#   624   625#   629   630#   634   635#   639   640#   644   645#
                          670    671#   675   676#   680   681#   685   686#   690   691#   715   716#   721
                          722#   726   727#   731   732#   736   737#   761   762#   767   768#   772   773#
                          777    778#   782   783#   808   809#   813   814#   818   819#   823   824#   828
                          829#   857   858#   862   863#   867   868#   872   873#   877   878#   882   883#
                          906    907#   914   915#   919   920#   924   925#   929   930#   934   935#   939
                          940#   944   945#   949   950#   954   955#   976   977#   985   986#   991   992#
                          996    997#  1001  1002#  1006  1007#  1011  1012#  1016  1017#  1021  1022#  1026
                         1027#  1050  1051#  1058  1059#  1063  1064#  1068  1069#  1073  1074#  1078  1079#
                         1083   1084#  1088  1089#  1093  1094#  1098  1099#  1120  1121#  1129  1130#  1135
                         1136#  1140  1141#  1145  1146#  1150  1151#  1155  1156#  1160  1161#  1165  1166#
                         1170   1171#  1196  1197#  1201  1202#  1206  1207#  1211  1212#  1216  1217#  1242
                         1243#  1247  1248#  1252  1253#  1257  1258#  1262  1263#  1292  1293#  1298  1299#
                         1303   1304#  1308  1309#  1313  1314#  1339  1340#  1344  1345#  1349  1350#  1354
                         1355#  1359  1360#  1389  1390#  1395  1396#  1400  1401#  1405  1406#  1410  1411#
                         1436   1437#  1441  1442#  1446  1447#  1451  1452#  1456  1457#  1482  1483#  1487
                         1488#  1492  1493#  1497  1498#  1502  1503#  1528  1529#  1533  1534#  1538  1539#
                         1543   1544#  1548  1549#  1574  1575#  1579  1580#  1584  1585#  1589  1590#  1594
                         1595#  1620  1621#  1625  1626#  1630  1631#  1635  1636#  1640  1641#  1666  1667#
                         1671   1672#  1676  1677#  1681  1682#  1686  1687#  1716  1717#  1722  1723#  1727
                         1728#  1732  1733#  1737  1738#  1759  1760#  1768  1769#  1774  1775#  1779  1780#
                         1784   1785#  1789  1790#  1794  1795#  1799  1800#  1804  1805#  1809  1810#  1833
                         1834#  1841  1842#  1846  1847#  1851  1852#  1856  1857#  1861  1862#  1866  1867#
                         1871   1872#  1876  1877#  1881  1882#  1907  1908#  1912  1913#  1917  1918#  1922
                         1923#  1927  1928#  1953  1954#  1958  1959#  1963  1964#  1968  1969#  1973  1974#
                         1999   2000#  2004  2005#  2009  2010#  2014  2015#  2019  2020#  2045  2046#  2050
                         2051#  2055  2056#  2060  2061#  2065  2066#  2090  2091#  2096  2097#  2101  2102#
                         2106   2107#  2111  2112#  2137  2138#  2142  2143#  2147  2148#  2152  2153#  2157
                         2158#  2186  2187#  2191  2192#  2196  2197#  2201  2202#  2206  2207#  2211  2212#
                         2240   2241#  2245  2246#  2250  2251#  2255  2256#  2260  2261#  2265  2266#  2289
                         2290#  2297  2298#  2302  2303#  2307  2308#  2312  2313#  2317  2318#  2322  2323#
                         2327   2328#  2332  2333#  2337  2338#  2359  2360#  2368  2369#  2374  2375#  2379
                         2380#  2384  2385#  2389  2390#  2394  2395#  2399  2400#  2404  2405#  2409  2410#
                         2435   2436#  2440  2441#  2445  2446#  2450  2451#  2455  2456#  2480  2481#  2486
```

CROSS REFERENCE TABLE -- USER SYMBOLS

```
                    2487#  2491   2492#  2496   2497#  2501   2502#  2527   2528#  2532   2533#  2537   2538#
                    2542   2543#  2547   2548#  2573   2574#  2578   2579#  2583   2584#  2588   2589#  2593
                    2594#  2622   2623#  2627   2628#  2632   2633#  2637   2638#  2642   2643#  2647   2648#
                    2671   2672#  2679   2680#  2684   2685#  2689   2690#  2694   2695#  2699   2700#  2704
                    2705#  2709   2710#  2714   2715#  2719   2720#  2743   2744#  2751   2752#  2756   2757#
                    2761   2762#  2766   2767#  2771   2772#  2776   2777#  2781   2782#  2786   2787#  2791
                    2792#  2815   2816#  2823   2824#  2828   2829#  2833   2834#  2838   2839#  2843   2844#
                    2848   2849#  2853   2854#  2858   2859#  2863   2864#  2885   2886#  2894   2895#  2900
                    2901#  2905   2906#  2915   2916#  2920   2921#  2925   2926#  2930   2931#  2935   2936#
                    2940   2941#  2982   2983#  2987   2988#  2993   2994#  2998   2999#  3003   3004#  3029
                    3030#  3037   3038#  3042   3043#  3047   3048#  3052   3053#  3057   3058#  3062   3063#
                    3067   3068#  3072   3073#  3076   3077#  3102   3103#  3110   3111#  3115   3116#  3120
                    3121#  3125   3126#  3130   3131#  3189   3190#  3196   3197#  3201   3202#  3206   3207#
                    3212   3213#  3217   3218#  3222   3223#  3227   3228#  3236   3237#  3295   3296#  3302
                    3303#  3307   3308#  3312   3313#  3318   3319#  3323   3324#  3328   3329#  3333   3334#
                    3342   3343#  3545   3546#
$FATAL  000402       106#  3557*
$GET42  015640      3372#
$HD   = 000003        14     15
$HIBTS  000430       143#
$ICNT   000503       174#  3401   3403   3405*  3410*  3575*
$ILLUP  017136      3632   3661#
$MAIL   000400       104#   144    148    203
$MBADR  000432       144#
$MSGAD  000414       111#
$MSGLG  000416       112#
$MSGTY  000400       105#  3567*
$PASS   000406       108#  3363*  3364*  3383
$PASTM  000436       146#
$POWER  017144      3659   3664#
$PSW    000432       152#   153    253    299    345    391    442    488    529    575    621    667    718
                      764   805    854    911    988   1055   1132   1193   1239   1295   1336   1392   1433
                     1479   1525   1571   1617   1663   1719   1771   1838   1904   1950   1996   2042   2093
                     2134   2183   2237   2294   2371   2432   2483   2524   2570   2619   2676   2748   2820
                     2897   2974*  2975*  2979   3034   3099*  3107   3193   3299   3438*  3439*  3453*  3486*
                     3487*  3499*  3531*  3532*
$PWRDN  017014       215   3632#  3656
$PWRMG  017132      3659#
$PWRUP  017062      3641   3646#
$SAVR6  017142      3640*  3646   3647*  3648*  3663#
$SETUP= 000020        82#  3363
$SP     000434       154#   155    252*   258    298*   304    344*   350    390*   396    528*   534    574*
                      580   620*   626    666*   672    804*   810    904*   910*   916   1048*  1054*  1060
                     1192*  1198   1238*  1244   1335*  1341   1432*  1438   1478*  1484   1524*  1530   1570*
                     1576   1616*  1622   1662*  1668   1831*  1837*  1843   1903*  1909   1949*  1955   1995*
                     2001   2041*  2047   2133*  2139   2287*  2293*  2299   2431*  2437   2523*  2529   2564*
                     2575   2669*  2675*  2681   2741*  2747*  2753   2813*  2819*  2825   2978*  2984   3027*
                     3033*  3039   3100*  3106*  3112   3186*  3192*  3198   3292*  3298*  3304   3443*  3464*
                     3537*
$STUP   = 177777      82#
$SVPC   = 001000      87#    92
$SWR    = 160000      14     15#   3356   3363   3373   3382   3383   3660
$SWREG  000422       116#  3147   3253   3393   3395   3397   3399   3550   3554   3569   3572
$TESTN  000404       107#   229*   273    319    365    411    457    503    549    595    641    687    733
                      779   825    879    951   1023   1095   1167   1213   1259   1310   1356   1407   1453
                     1499   1545   1591   1637   1683   1734   1806   1878   1924   1970   2016   2062   2108
```

CROSS REFERENCE TABLE -- USER SYMBOLS

```
                    2154   2208   2262   2334   2406   2452   2498   2544   2590   2644   2716   2788   2860
                    2937   3000   3073   3127   3233   3339   3395   3406*
$TN    = 000001       14#
$TPB     000552      193#    225   3170*  3173*  3276*  3279*  3688*
$TPS     000550      192#   3159*  3165*  3171   3174*  3265*  3271*  3277   3280*  3686
$TSTM    000434      145#
$TYPE    017154      214    3675#
$UNIT    000412      110#
$UNITM   000440      147#
$USWR    000424      117#
.      = 017224       81#     82     87     88#    90#    92#    96#   132    133#   135#   137#   149#   151#
                     153#   155#   157#   159#   161#   199#   210#   254    259    264    269    274    300
                     305    310    315    320    346    351    356    361    366    392    397    402    407
                     412    443    448    453    458    489    494    499    504    530    535    540    545
                     550    576    581    586    591    596    622    627    632    637    642    668    673
                     678    683    688    719    724    729    734    765    770    775    780    806    811
                     816    821    826    855    860    865    870    875    880    912    917    922    927
                     932    937    942    947    952    989    994    999   1004   1009   1014   1019   1024
                    1056   1061   1066   1071   1076   1081   1086   1091   1096   1133   1138   1143   1148
                    1153   1158   1163   1168   1194   1199   1204   1209   1214   1240   1245   1250   1255
                    1260   1296   1301   1306   1311   1337   1342   1347   1352   1357   1393   1398   1403
                    1408   1434   1439   1444   1449   1454   1480   1485   1490   1495   1500   1526   1531
                    1536   1541   1546   1572   1577   1582   1587   1592   1618   1623   1628   1633   1638
                    1664   1669   1674   1679   1684   1720   1725   1730   1735   1772   1777   1782   1787
                    1792   1797   1802   1807   1839   1844   1849   1854   1859   1864   1869   1874   1879
                    1905   1910   1915   1920   1925   1951   1956   1961   1966   1971   1997   2002   2007
                    2012   2017   2043   2048   2053   2058   2063   2094   2099   2104   2109   2135   2140
                    2145   2150   2155   2184   2189   2194   2199   2204   2209   2238   2243   2248   2253
                    2258   2263   2295   2300   2305   2310   2315   2320   2325   2330   2335   2372   2377
                    2382   2387   2392   2397   2402   2407   2433   2438   2443   2448   2453   2484   2489
                    2494   2499   2525   2530   2535   2540   2545   2571   2576   2581   2586   2591   2620
                    2625   2630   2635   2640   2645   2677   2682   2687   2692   2697   2702   2707   2712
                    2717   2749   2754   2759   2764   2769   2774   2779   2784   2789   2821   2826   2831
                    2836   2841   2846   2851   2856   2861   2898   2903   2913   2918   2923   2928   2933
                    2938   2965#  2966#  2974#  2975#  2980   2985   2991   2996   3001   3020#  3021#  3035
                    3040   3045   3050   3055   3060   3065   3070   3074   3092#  3093#  3099#  3100#  3108
                    3113   3118   3123   3128   3151   3194   3199   3204   3210   3215   3220   3225   3234
                    3238#  3239#  3257   3300   3305   3310   3316   3321   3326   3331   3340   3344#  3345#
                    3383   3386#  3388#  3389#  3427#  3428#  3438#  3439#  3453#  3454#  3475#  3476#  3486#
                    3487#  3499#  3500#  3521#  3522#  3531#  3532#  3568   3570   3573   3596   3611   3614
                    3619   3626#  3643   3662
.BIT   = 177777        4#     35     36     37     38     39     40     41     42   3550   3554   3572
.PR      000544      190#   3595*  3597*  3598*  3612*  3613   3616*
.PRBUF   017004     3600   3618   3622   3626#
.PRF     016724     3602   3608#
.PRL     016712     3603#  3617
.$X    = 000430      132#   137
..A    = 016374     2965#  2974#  3020#  3092#  3099#  3238#  3344#  3427#  3438#  3453#  3475#  3486#  3499#
                    3521#  3531#
..B    = 016400     2965#  2966   2974#  2975   3020#  3021   3092#  3093   3099#  3100   3238#  3239   3344#
                    3345   3427#  3428   3438#  3439   3453#  3454   3475#  3476   3486#  3487   3499#  3500
                    3521#  3522   3531#  3532
..C    = 000067     2965#  2974#  3020#  3092#  3099#  3238#  3344#  3427#  3438#  3453#  3475#  3486#  3499#
                    3521#  3531#
```

```
DVKACB  MACY11 27(1006)  17-JAN-77  15:14  PAGE 101                                SEQ 0108
DVKACB.P11    13-JAN-77 15:28          CROSS REFERENCE TABLE -- USER SYMBOLS

. ABS.  017224     000


 ERRORS DETECTED:  0
 DEFAULT GLOBALS GENERATED:  0

 DVKACB,DVKACB/SOL/CRF=DVKACB.SML,DVKACB.P11
 RUN-TIME: 47 62 4 SECONDS
 RUN-TIME RATIO: 267/114=2.3
 CORE USED:  41K  (82 PAGES)
```