LSI11

FIS DIAGNOSTIC

MD-11-DVKAC-A

EP-DVKAC-A-DL-A

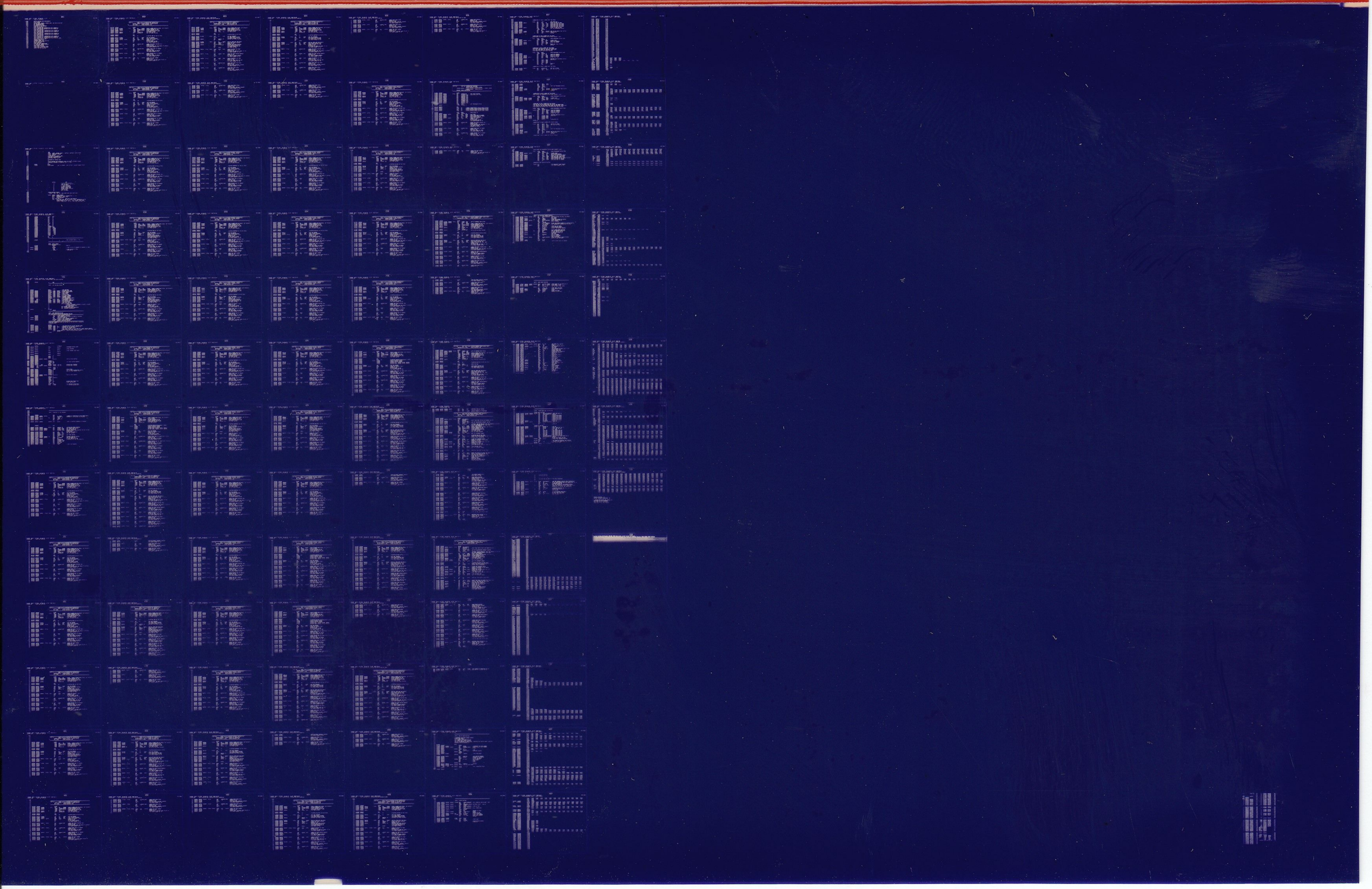
COPYRIGHT © 1976

FICHE 1 OF 1

OCT 1976

digital

MADE IN USA



5046

```
                                .ABS
                                .MCALL  .SEOP,.HEADER,.SACT11,.SAPTBLS,.SAPTHDR,.SETUP,STARS
                                .MCALL  PUSH,POP,.SPOWER
                                .TITLE  DVKACA
                                ;#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
                000001                  R1=     %1
                000002                  R2=     %2
                000003                  R3=     %3
                000004                  R4=     %4
                000005                  R5=     %5
                000005                  TTY=    %5
                000006                  SP=     %6
                000007                  PC=     %7
                000024                  PWRVEC= 24
                104400                  SCOPE=  TRAP
                100000                  SW15=   100000
                040000                  SW14=    40000
                020000                  SW13=    20000
                010000                  SW12=    10000
                004000                  SW11=     4000
                002000                  SW10=     2000
                001000                  SW09=     1000
                000400                  SW08=      400
                000004                  TYPE=   IOT
                000001                  N=      1
                000001                  SF=     1
                                        ;*****************************************************************
                000000                  .=      0                       ;TRAP CATCHER FROM 0 - 776
                                        ;*****************************************************************
(1)
(1)                                     .SBTTL  ACT11 HOOKS
(1)                                     ;HOOKS REQUIRED BY ACT11
(1)             001000                          $SVPC=.                 ;SAVE PC
(1)             000046                          .=46
(1)     000046  015042                  $ENDAD                          ;;1)SET LOC.46 TO ADDRESS OF $ENDAD IN .SEOP
(1)             000052                          .=52
(1)     000052  000000                  .WORD   0                       ;;2)SET LOC.52 TO ZERO
(1)             001000                          .=$SVPC                 ;; RESTORE PC
5331
5332
```

```
                        .=      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                          .=      $APTHD
000430                  HLTAD$:
```

```
5343                    000432                          .=      HLTAD$+2
5344    000432          000432  SPSW:
5345                    000434                          .=      SPSW+2
5346    000434          000434  SSP:                                    ;PROCESSOR STATUS WORD
5347                    000436                          .=      SSP+2
5348    000436          000436  ANS1:                                   ;STACK POINTER
5349                    000440                          .=      ANS1+2
5350    000440          000440  ANS2:                                   ;FIRST ANSWER (SEE CODE)
5351                    000442                          .=      ANS2+2
5352    000442          000442  ANS3:
5353                    000444                          .=      ANS3+2
5354    000444  000000  000444  ANS4:   0
5355    000446  000000  000446  ANS5:   0
5356    000450  000000  000450  ANS6:   0
5357    000452  000000  000452          0,0,0,0                         ;NON-%6 STACK BUFFER
        000460  000000  000460
5358    000462  000000  000462  ERRORS: 0
5359    000464  000244  000464  FISVEC: 244                             ;FIS TRAP VECTOR ADDRESS
5360    000466  000246  000466  FISLVL: 246
5361    000470  000000  000470  LAD$:   0
5362    000472  006412  000472  RETURN: .ASCIZ  <12><15>                ;RETURN AND LINEFEED
5363    000475     015  000475  SPACE:  .ASCIZ  <15><12>"   "           ;RETURN AND 3 SPACES
        000502     000  000502
5364    000503     000  000503  $ICNT:  .BYTE   0
5365                    000504          .EVEN
5366    000504  000007  000506  $BELL:  .WORD   7                       ;RING A BELL
5367    000506  000000  000510  SAVTPS: 0                               ;LOC TO SAVE TELEPRINTER STATUS
5368    000510  000000  000512  STACK0: 0                               ;NON-%6 STACK NORMAL LIMIT
5369    000512  000000  000514  STACK2: 0                               ;
5370    000514  000000  000516  STACK4: 0                               ;
5371    000516  000000  000520  STACK6: 0                               ;
5372    000520  000000  000520          0,0,0,0,0                       ;NON-%6 STACK BUFFER
        000526  000000  000526
5373    000532  000000  000532  STAK10: 0
5374                    000511          STACK1 = STACK0+1
5375    000534  000000  000534  TEMP:   0
5376    000536  000000  000536  TIMES:  0
5377    000540  000000  000540  TYPCNT: 0
5378    000542  000006  000542  YESRT:  RTT                             ;RETURN FROM TRACE TRAP
5379    000544  000000  000544  .PR:    0                               ;COUNT AND SWITCH
5380    000546  000064  000546  TTYOUT: 64
5381    000550  177564  000550  $TPS:   177564                          ;TTY PRINTER STATUS REG.
5382    000552  177566  000552  $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  016346  000020          MOV     #$TYPE, @#20    ;SET UP VECTOR 20
000620  012737  016206  000024          MOV     #$PWRDN ,@#24   ;SERVICE POWER DOWN ROUTINE FOR ANY FUTURE
                                                        ;POWER DOWN
000626  012700  000030          MOV     #30,    R0      ;SET R0 TO VECTOR 30
000632  012720  015642          MOV     #HLTS,  (0)+    ;SET EMT VECTOR
000636  012720  000340          MOV     #340,   (0)+
000642  012720  015074          MOV     #SCOPES,(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     #STPB+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     $TESTN
000714  005067  177550          CLR     LAD$            ;CLEAR LOOP ADDRESS
```

```
7147
 (1)                                ;*************************************************************************
 (1)                                ;TEST 1:        FADD (LSI-11 FLOATING ADD INSTRUCTION)
 (1)                                ;       000000,000000 + 000000,000000 = 000000,000000
 (1)                                ;       PS = 004,       STACK POINTER = R0
 (1)                                ;*************************************************************************
 (1)
 (1)  000720  104400                            SCOPE
 (1)  000722  004567  014436        TST1:       JSR     R5,     PUSHR   ;PUSH 4 WORDS ONTO R0 STACK, SET PRIORITY
 (1)  000726  000000  000000                    .WORD   000000,000000   ;SECOND OPERAND ON TOP
 (1)  000732  000000  000000                    .WORD   000000,000000   ;FIRST OPERAND ON BOTTOM
 (1)  000736  000000                            .WORD   000             ;PROCESSOR PRIORITY LEVEL
 (1)  000740  015634  000340                    .WORD   TRAPER,340      ;FIS TRAP VECTOR
 (1)  000744  012700  000510                    MOV     #STACK0,R0      ;SET UP STACK POINTER
 (1)
 (1)  000750  000240                            NOP
 (1)  000752  075000                            FADD    R0              ;FLOATING ADD ON THE R0 STACK
 (1)
 (1)  000754  004767  014436                    JSR     PC,     POPR    ;POP THE ANSWER
 (1)  000760  010067  177450                    MOV     R0,     $SP     ;SAVE "STACK POINTER"
 (1)  000764  122767  000004  177440           CMPB    #004,   $PSW    ;CHECK PS (EXCEPT T BIT)
 (1)  000772  001402                            BEQ     .+6             ;BRANCH IF OK
 (1)  000774  104000                            HLT                     ;PS NOT EQUAL TO 004
 (3)  000776  000001                            1                       ;THE ERROR NUMBER IS 1
 (1)
 (1)  001000  022767  000514  177426            CMP     #STACK4,$SP     ;CHECK THE STACK POINTER (R0)
 (1)  001006  001402                            BEQ     .+6             ;BRANCH IF OK
 (1)  001010  104000                            HLT                     ;STACK POINTER (R0) NOT EQUAL TO #STACK4
 (3)  001012  000002                            2                       ;THE ERROR NUMBER IS 2
 (1)
 (1)  001014  005767  177416                    TST     ANS1            ;CHECK FIRST HALF OF ANSWER
 (1)  001020  001402                            BEQ     .+6             ;BRANCH IF OK
 (1)  001022  104002                            HLT+2                   ;ANS1 NOT EQUAL TO 000000
 (3)  001024  000003                            3                       ;THE ERROR NUMBER IS 3
 (1)
 (1)  001026  005767  177406                    TST     ANS2            ;CHECK SECOND HALF OF ANSWER
 (1)  001032  001402                            BEQ     .+6             ;BRANCH IF OK
 (1)  001034  104002                            HLT+2                   ;ANS2 NOT EQUAL TO 000000
 (3)  001036  000004                            4                       ;THE ERROR NUMBER IS 4
 (1)
 (1)  001040  122767  000001  177336  END1:     CMPB    #1,     $TESTN  ;CHECK THE TEST NUMBER
 (1)  001046  001402                            BEQ     .+6             ;BRANCH IF OK
 (1)  001050  104000                            HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
 (3)  001052  000005                            5                       ;THE ERROR NUMBER IS 5
 (1)
 (1)
```

```
7148
 (1)                                ;*****************************************************************************
 (1)                                ;TEST 2:        FADD (LSI-11 FLOATING ADD INSTRUCTION)
 (1)                                ;       040200,000000 + 040200,000000 = 040400,000000
 (1)                                ;       PS = 000,       STACK POINTER = R1
 (1)                                ;*****************************************************************************
 (1)
 (1)  001054  104400                            SCOPE
 (1)  001056  004567  014302        TST2:       JSR     R5,     PUSHR   ;PUSH 4 WORDS ONTO R1 STACK, SET PRIORITY
 (1)  001062  040200  000000                    .WORD   040200,000000   ;SECOND OPERAND ON TOP
 (1)  001066  040200  000000                    .WORD   040200,000000   ;FIRST OPERAND ON BOTTOM
 (1)  001072  000040                            .WORD   040             ;PROCESSOR PRIORITY LEVEL
 (1)  001074  015634  000340                    .WORD   TRAPER,340      ;FIS TRAP VECTOR
 (1)  001100  012701  000510                    MOV     #STACK0,R1      ;SET UP STACK POINTER
 (1)
 (1)  001104  000240                            NOP
 (1)  001106  075001                            FADD    R1              ;FLOATING ADD ON THE R1 STACK
 (1)
 (1)  001110  004767  014302                    JSR     PC,     POPR    ;POP THE ANSWER
 (1)  001114  010167  177314                    MOV     R1,     $SP     ;SAVE "STACK POINTER"
 (1)  001120  105767  177306                    TSTB    $PSW            ;CHECK PS (EXCEPT T BIT)
 (1)  001124  001402                            BEQ     .+6             ;BRANCH IF OK
 (1)  001126  104000                            HLT                     ;PS NOT EQUAL TO 000
 (3)  001130  000006                            6                       ;THE ERROR NUMBER IS 6
 (1)
 (1)  001132  022767  000514  177274            CMP     #STACK4,$SP     ;CHECK THE STACK POINTER (R1)
 (1)  001140  001402                            BEQ     .+6             ;BRANCH IF OK
 (1)  001142  104000                            HLT                     ;STACK POINTER (R1) NOT EQUAL TO #STACK4
 (3)  001144  000007                            7                       ;THE ERROR NUMBER IS 7
 (1)
 (1)  001146  022767  040400  177262            CMP     #040400,ANS1    ;CHECK FIRST HALF OF ANSWER
 (1)  001154  001402                            BEQ     .+6             ;BRANCH IF OK
 (1)  001156  104002                            HLT+2                   ;ANS1 NOT EQUAL TO 040400
 (3)  001160  000010                            10                      ;THE ERROR NUMBER IS 10
 (1)
 (1)  001162  005767  177252                    TST     ANS2            ;CHECK SECOND HALF OF ANSWER
 (1)  001166  001402                            BEQ     .+6             ;BRANCH IF OK
 (1)  001170  104002                            HLT+2                   ;ANS2 NOT EQUAL TO 000000
 (3)  001172  000011                            11                      ;THE ERROR NUMBER IS 11
 (1)
 (1)  001174  122767  000002  177202  END2:     CMPB    #2,     $TESTN  ;CHECK THE TEST NUMBER
 (1)  001202  001402                            BEQ     .+6             ;BRANCH IF OK
 (1)  001204  104000                            HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
 (3)  001206  000012                            12                      ;THE ERROR NUMBER IS 12
 (1)
 (1)
```

```
7149
 (1)                                ;*************************************************************************
 (1)                                ;TEST 3:       FADD (LSI-11 FLOATING ADD INSTRUCTION)
 (1)                                ;       077777,177777 + 177777,177777 = 000000,000000
 (1)                                ;       PS = 004,       STACK POINTER = R2
 (1)                                ;*************************************************************************
 (1)
 (1)  001210  104400                          SCOPE
 (1)  001212  004567  014146          TST3:   JSR     R5,     PUSHR   ;PUSH 4 WORDS ONTO R2 STACK, SET PRIORITY
 (1)  001216  177777  177777                  .WORD   177777,177777   ;SECOND OPERAND ON TOP
 (1)  001222  077777  177777                  .WORD   077777,177777   ;FIRST OPERAND ON BOTTOM
 (1)  001226  000100                          .WORD   100             ;PROCESSOR PRIORITY LEVEL
 (1)  001230  015634  000340                  .WORD   TRAPER,340      ;FIS TRAP VECTOR
 (1)  001234  012702  000510                  MOV     #STACK0,R2      ;SET UP STACK POINTER
 (1)
 (1)  001240  070240                          NOP
 (1)  001242  075002                          FADD    R2              ;FLOATING ADD ON THE R2 STACK
 (1)
 (1)  001244  004767  014146                  JSR     PC,     POPR    ;POP THE ANSWER
 (1)  001250  010267  177160                  MOV     R2,     $SP     ;SAVE "STACK POINTER"
 (1)  001254  122767  000004  177150          CMPB    #004,   $PSW    ;CHECK PS (EXCEPT T BIT)
 (1)  001262  001402                          BEQ     .+6             ;BRANCH IF OK
 (1)  001264  104000                          HLT                     ;PS NOT EQUAL TO 004
 (3)  001266  000013                          13                      ;THE ERROR NUMBER IS 13
 (1)
 (1)  001270  022767  000514  177136          CMP     #STACK4,$SP     ;CHECK THE STACK POINTER (R2)
 (1)  001276  001402                          BEQ     .+6             ;BRANCH IF OK
 (1)  001300  104000                          HLT                     ;STACK POINTER (R2) NOT EQUAL TO #STACK4
 (3)  001302  000014                          14                      ;THE ERROR NUMBER IS 14
 (1)
 (1)  001304  005767  177126                  TST     ANS1            ;CHECK FIRST HALF OF ANSWER
 (1)  001310  001402                          BEQ     .+6             ;BRANCH IF OK
 (1)  001312  104002                          HLT+2                   ;ANS1 NOT EQUAL TO 000000
 (3)  001314  000015                          15                      ;THE ERROR NUMBER IS 15
 (1)
 (1)  001316  005767  177116                  TST     ANS2            ;CHECK SECOND HALF OF ANSWER
 (1)  001322  001402                          BEQ     .+6             ;BRANCH IF OK
 (1)  001324  104002                          HLT+2                   ;ANS2 NOT EQUAL TO 000000
 (3)  001326  000016                          16                      ;THE ERROR NUMBER IS 16
 (1)
 (1)  001330  122767  000003  177046  END3:   CMPB    #3,     $TESTN  ;CHECK THE TEST NUMBER
 (1)  001336  001402                          BEQ     .+6             ;BRANCH IF OK
 (1)  001340  104000                          HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
 (3)  001342  000017                          17                      ;THE ERROR NUMBER IS 17
 (1)
 (1)
```

```
7150
 (1)
 (1)                                ;*****************************************************************************
 (1)                                ;TEST 4:        FADD (LSI-11 FLOATING ADD INSTRUCTION)
 (1)                                ;       052525,052525 + 152525,052524 = 044600,000000
 (1)                                ;       PS = 200,       STACK POINTER = SP
 (1)                                ;*****************************************************************************
 (1)
 (1)  001344  104400                                SCOPE
 (1)  001346  004567  013640                        TST4:   JSR     R5,     PUSHS   ;PUSH 4 WORDS ONTO STACK, SET PRIORITY
 (1)  001352  152525  052524                        .WORD   152525,052524   ;SECOND OPERAND ON TOP
 (1)  001356  052525  052525                        .WORD   052525,052525   ;FIRST OPERAND ON BOTTOM
 (1)  001362  000217                                .WORD   217             ;PROCESSOR PRIORITY LEVEL
 (1)  001364  015634  000340                        .WORD   TRAPER,340      ;FIS TRAP VECTOR
 (1)
 (1)  001370  000240                                NOP
 (1)  001372  075006                                FADD    SP              ;FLOATING ADD ON THE STACK
 (1)
 (1)  001374  004767  013652                        JSR     PC,     POPS    ;POP THE ANSWER
 (1)  001400  022706  000600                        CMP     #BEGIN, SP      ;CHECK THE STACK POINTER
 (1)  001404  001405                                BEQ     TSA4            ;BRANCH IF OK
 (1)  001406  012706  000600                        MOV     #BEGIN, SP      ;RESTORE STACK POINTER
 (1)  001412  104000                                HLT                     ;STACK POINTER FOULED UP
 (3)  001414  000020                                20                      ;THE ERROR NUMBER IS 20
 (1)  001416  000421                                BR      END4            ;SKIP REST OF TEST
 (1)
 (1)  001420  122767  000200  177004  TSA4:         CMPB    #200,   SPSW    ;CHECK PS (EXCEPT T BIT)
 (1)  001426  001402                                BEQ     .+6             ;BRANCH IF OK
 (1)  001430  104000                                HLT                     ;PS NOT EQUAL TO 200
 (3)  001432  000021                                21                      ;THE ERROR NUMBER IS 21
 (1)
 (1)  001434  022767  044600  176774                CMP     #044600,ANS1    ;CHECK FIRST HALF OF ANSWER
 (1)  001442  001402                                BEQ     .+6             ;BRANCH IF OK
 (1)  001444  104002                                HLT+2                   ;ANS1 NOT EQUAL TO 044600
 (3)  001446  000022                                22                      ;THE ERROR NUMBER IS 22
 (1)
 (1)  001450  005767  176764                        TST     ANS2            ;CHECK SECOND HALF OF ANSWER
 (1)  001454  001402                                BEQ     .+6             ;BRANCH IF OK
 (1)  001456  104002                                HLT+2                   ;ANS2 NOT EQUAL TO 000000
 (3)  001460  000023                                23                      ;THE ERROR NUMBER IS 23
 (1)
 (1)  001462  122767  000004  176714  END4:         CMPB    #4,     STESTN  ;CHECK THE TEST NUMBER
 (1)  001470  001402                                BEQ     .+6             ;BRANCH IF OK
 (1)  001472  104000                                HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
 (3)  001474  000024                                24                      ;THE ERROR NUMBER IS 24
 (1)
 (1)
```

```
7151
 (1)                                ;*************************************************************************
 (1)                                ;TEST 5:        FADD (LSI-11 FLOATING ADD INSTRUCTION)
 (1)                                ;       125200,000000 + 025177,177777 = 117200,000000
 (1)                                ;       PS = 210,       STACK POINTER = SP
 (1)                                ;*************************************************************************
 (1)
 (1)  001476  104400                        SCOPE
 (1)  001500  004567  013506                TST5:   JSR     R5,     PUSHS   ;PUSH 4 WORDS ONTO STACK, SET PRIORITY
 (1)  001504  025177  177777                .WORD   025177,177777   ;SECOND OPERAND ON TOP
 (1)  001510  125200  000000                .WORD   125200,000000   ;FIRST OPERAND ON BOTTOM
 (1)  001514  000307                        .WORD   307             ;PROCESSOR PRIORITY LEVEL
 (1)  001516  015634  000340                .WORD   TRAPER,340      ;FIS TRAP VECTOR
 (1)
 (1)  001522  000240                        NOP
 (1)  001524  075006                        FADD    SP              ;FLOATING ADD ON THE STACK
 (1)
 (1)  001526  004767  013520                JSR     PC,     POPS    ;POP THE ANSWER
 (1)  001532  022706  000600                CMP     #BEGIN, SP      ;CHECK THE STACK POINTER
 (1)  001536  001405                        BEQ     TSA5            ;BRANCH IF OK
 (1)  001540  012706  000600                MOV     #BEGIN, SP      ;RESTORE STACK POINTER
 (1)  001544  104000                        HLT                     ;STACK POINTER FOULED UP
 (3)  001546  000025                        25                      ;THE ERROR NUMBER IS 25
 (1)  001550  000421                        BR      ENDS            ;SKIP REST OF TEST
 (1)
 (1)  001552  122767  000210  176652  TSA5:  CMPB    #210,   $PSW    ;CHECK PS (EXCEPT T BIT)
 (1)  001560  001402                        BEQ     .+6             ;BRANCH IF OK
 (1)  001562  104000                        HLT                     ;PS NOT EQUAL TO 210
 (3)  001564  000026                        26                      ;THE ERROR NUMBER IS 26
 (1)
 (1)  001566  022767  117200  176642        CMP     #117200,ANS1    ;CHECK FIRST HALF OF ANSWER
 (1)  001574  001402                        BEQ     .+6             ;BRANCH IF OK
 (1)  001576  104002                        HLT+2                   ;ANS1 NOT EQUAL TO 117200
 (3)  001600  000027                        27                      ;THE ERROR NUMBER IS 27
 (1)
 (1)  001602  005767  176632                TST     ANS2            ;CHECK SECOND HALF OF ANSWER
 (1)  001606  001402                        BEQ     .+6             ;BRANCH IF OK
 (1)  001610  104002                        HLT+2                   ;ANS2 NOT EQUAL TO 000000
 (3)  001612  000030                        30                      ;THE ERROR NUMBER IS 30
 (1)
 (1)  001614  122767  000005  176562  ENDS:  CMPB    #5,     $TESTN  ;CHECK THE TEST NUMBER
 (1)  001622  001402                        BEQ     .+6             ;BRANCH IF OK
 (1)  001624  104000                        HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
 (3)  001626  000031                        31                      ;THE ERROR NUMBER IS 31
 (1)
 (1)
```

```
7152
 (1)                                ;*************************************************************************
 (1)                                ;TEST 6:        FADD (LSI-11 FLOATING ADD INSTRUCTION)
 (1)                                ;       135753,024642 + 100125,052525 = 135753,024642
 (1)                                ;       PS = 210,       STACK POINTER = R5
 (1)                                ;*************************************************************************
 (1)
 (1)  001630  104400                            SCOPE
 (1)  001632  004567  013526        TST6:  JSR     R5,     PUSHR   ;PUSH 4 WORDS ONTO R5 STACK, SET PRIORITY
 (1)  001636  100125  052525               .WORD   100125,052525   ;SECOND OPERAND ON TOP
 (1)  001642  135753  024642               .WORD   135753,024642   ;FIRST OPERAND ON BOTTOM
 (1)  001646  000347                       .WORD   347             ;PROCESSOR PRIORITY LEVEL
 (1)  001650  015634  000340               .WORD   TRAPER,340      ;FIS TRAP VECTOR
 (1)  001654  012705  000510               MOV     #STACK0,R5      ;SET UP STACK POINTER
 (1)
 (1)  001660  000240                       NOP
 (1)  001662  075005                       FADD    R5              ;FLOATING ADD ON THE R5 STACK
 (1)
 (1)  001664  004767  013526               JSR     PC,     POPR    ;POP THE ANSWER
 (1)  001670  010567  176540               MOV     R5,     $SP     ;SAVE "STACK POINTER"
 (1)  001674  122767  000210  176530       CMPB    #210,   $PSW    ;CHECK PS (EXCEPT T BIT)
 (1)  001702  001402                       BEQ     .+6             ;BRANCH IF OK
 (1)  001704  104000                       HLT                     ;PS NOT EQUAL TO 210
 (3)  001706  000032                       32                      ;THE ERROR NUMBER IS 32
 (1)
 (1)  001710  022767  000514  176516       CMP     #STACK4,$SP     ;CHECK THE STACK POINTER (R5)
 (1)  001716  001402                       BEQ     .+6             ;BRANCH IF OK
 (1)  001720  104000                       HLT                     ;STACK POINTER (R5) NOT EQUAL TO #STACK4
 (3)  001722  000033                       33                      ;THE ERROR NUMBER IS 33
 (1)
 (1)  001724  022767  135753  176504       CMP     #135753,ANS1    ;CHECK FIRST HALF OF ANSWER
 (1)  001732  001402                       BEQ     .+6             ;BRANCH IF OK
 (1)  001734  104002                       HLT+2                   ;ANS1 NOT EQUAL TO 135753
 (3)  001736  000034                       34                      ;THE ERROR NUMBER IS 34
 (1)
 (1)  001740  022767  024642  176472       CMP     #024642,ANS2    ;CHECK SECOND HALF OF ANSWER
 (1)  001746  001402                       BEQ     .+6             ;BRANCH IF OK
 (1)  001750  104002                       HLT+2                   ;ANS2 NOT EQUAL TO 024642
 (3)  001752  000035                       35                      ;THE ERROR NUMBER IS 35
 (1)
 (1)  001754  122767  000006  176422  END6:CMPB    #6,     $TESTN  ;CHECK THE TEST NUMBER
 (1)  001762  001402                       BEQ     .+6             ;BRANCH IF OK
 (1)  001764  104000                       HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
 (3)  001766  000036                       36                      ;THE ERROR NUMBER IS 36
 (1)
 (1)
```

```
7153
 (1)                                ;*************************************************************************
 (1)                                ;TEST 7:        FADD (LSI-11 FLOATING ADD INSTRUCTION)
 (1)                                ;       000052,125252 + 001357,024642 = 001357,024642
 (1)                                ;       PS = 200,       STACK POINTER = R1
 (1)                                ;*************************************************************************
 (1)
 (1) 001770  104400                         SCOPE
 (1) 001772  004567  013366         TST7:   JSR     R5,     PUSHR   ;PUSH 4 WORDS ONTO R1 STACK, SET PRIORITY
 (1) 001776  001357  024642                 .WORD   001357,024642   ;SECOND OPERAND ON TOP
 (1) 002002  000052  125252                 .WORD   000052,125252   ;FIRST OPERAND ON BOTTOM
 (1) 002006  000257                         .WORD   257             ;PROCESSOR PRIORITY LEVEL
 (1) 002010  015634  000340                 .WORD   TRAPER,340      ;FIS TRAP VECTOR
 (1) 002014  012701  000510                 MOV     #STACK0,R1      ;SET UP STACK POINTER
 (1)
 (1) 002020  000240                         NOP
 (1) 002022  075001                         FADD    R1              ;FLOATING ADD ON THE R1 STACK
 (1)
 (1) 002024  004767  013366                 JSR     PC,     POPR    ;POP THE ANSWER
 (1) 002030  010167  176400                 MOV     R1,     $SP     ;SAVE "STACK POINTER"
 (1) 002034  122767  000200  176370         CMPB    #200,   $PSW    ;CHECK PS (EXCEPT T BIT)
 (1) 002042  001402                         BEQ     .+6             ;BRANCH IF OK
 (1) 002044  104000                         HLT                     ;PS NOT EQUAL TO 200
 (3) 002046  000037                         37                      ;THE ERROR NUMBER IS 37
 (1)
 (1) 002050  022767  000514  176356         CMP     #STACK4,$SP     ;CHECK THE STACK POINTER (R1)
 (1) 002056  001402                         BEQ     .+6             ;BRANCH IF OK
 (1) 002060  104000                         HLT                     ;STACK POINTER (R1) NOT EQUAL TO #STACK4
 (3) 002062  000040                         40                      ;THE ERROR NUMBER IS 40
 (1)
 (1) 002064  022767  001357  176344         CMP     #001357,ANS1    ;CHECK FIRST HALF OF ANSWER
 (1) 002072  001402                         BEQ     .+6             ;BRANCH IF OK
 (1) 002074  104002                         HLT+2                   ;ANS1 NOT EQUAL TO 001357
 (3) 002076  000041                         41                      ;THE ERROR NUMBER IS 41
 (1)
 (1) 002100  022767  024642  176332         CMP     #024642,ANS2    ;CHECK SECOND HALF OF ANSWER
 (1) 002106  001402                         BEQ     .+6             ;BRANCH IF OK
 (1) 002110  104002                         HLT+2                   ;ANS2 NOT EQUAL TO 024642
 (3) 002112  000042                         42                      ;THE ERROR NUMBER IS 42
 (1)
 (1) 002114  122767  000007  176262 END7:   CMPB    #7,     $TESTN  ;CHECK THE TEST NUMBER
 (1) 002122  001402                         BEQ     .+6             ;BRANCH IF OK
 (1) 002124  104000                         HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
 (3) 002126  000043                         43                      ;THE ERROR NUMBER IS 43
 (1)
 (1)
```

```
7154
 (1)                                ;*************************************************************************
 (1)                                ;TEST 10:       FADD (LSI-11 FLOATING ADD INSTRUCTION)
 (1)                                ;       100400,000000 + 000200,000000 = 100200,000000
 (1)                                ;       PS = 010,       STACK POINTER = R5
 (1)                                ;*************************************************************************
 (1)
 (1) 002130  104400                         SCOPE
 (1) 002132  004567  013226         TST10:  JSR     R5,     PUSHR   ;PUSH 4 WORDS ONTO R5 STACK, SET PRIORITY
 (1) 002136  000200  000000                 .WORD   000200,000000   ;SECOND OPERAND ON TOP
 (1) 002142  100400  000000                 .WORD   100400,000000   ;FIRST OPERAND ON BOTTOM
 (1) 002146  000140                         .WORD   140             ;PROCESSOR PRIORITY LEVEL
 (1) 002150  015634  000340                 .WORD   TRAPER,340      ;FIS TRAP VECTOR
 (1) 002154  012705  000510                 MOV     #STACK0,R5      ;SET UP STACK POINTER
 (1)
 (1) 002160  000240                         NOP
 (1) 002162  075005                         FADD    R5              ;FLOATING ADD ON THE R5 STACK
 (1)
 (1) 002164  004767  013226                 JSR     PC,     POPR    ;POP THE ANSWER
 (1) 002170  010567  176240                 MOV     R5,     $SP     ;SAVE "STACK POINTER"
 (1) 002174  122767  000010  176230         CMPB    #010,   $PSW    ;CHECK PS (EXCEPT T BIT)
 (1) 002202  001402                         BEQ     .+6             ;BRANCH IF OK
 (1) 002204  104000                         HLT                     ;PS NOT EQUAL TO 010
 (3) 002206  000044                         44                      ;THE ERROR NUMBER IS 44
 (1)
 (1) 002210  022767  000514  176216         CMP     #STACK4,$SP     ;CHECK THE STACK POINTER (R5)
 (1) 002216  001402                         BEQ     .+6             ;BRANCH IF OK
 (1) 002220  104000                         HLT                     ;STACK POINTER (R5) NOT EQUAL TO #STACK4
 (3) 002222  000045                         45                      ;THE ERROR NUMBER IS 45
 (1)
 (1) 002224  022767  100200  176204         CMP     #100200,ANS1    ;CHECK FIRST HALF OF ANSWER
 (1) 002232  001402                         BEQ     .+6             ;BRANCH IF OK
 (1) 002234  104002                         HLT+2                   ;ANS1 NOT EQUAL TO 100200
 (3) 002236  000046                         46                      ;THE ERROR NUMBER IS 46
 (1)
 (1) 002240  005767  176174                 TST     ANS2            ;CHECK SECOND HALF OF ANSWER
 (1) 002244  001402                         BEQ     .+6             ;BRANCH IF OK
 (1) 002246  104002                         HLT+2                   ;ANS2 NOT EQUAL TO 000000
 (3) 002250  000047                         47                      ;THE ERROR NUMBER IS 47
 (1)
 (1) 002252  122767  000010  176124 END10:  CMPB    #10,    $TESTN  ;CHECK THE TEST NUMBER
 (1) 002260  001402                         BEQ     .+6             ;BRANCH IF OK
 (1) 002262  104000                         HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
 (3) 002264  000050                         50                      ;THE ERROR NUMBER IS 50
 (1)
 (1)
```

```
7155
 (1)
 (1)                                    ;*************************************************************************
 (1)                                    ;TEST 11:       FADD (LSI-11 FLOATING ADD INSTRUCTION)
 (1)                                    ;       000425,052525 + 100252,125252 = 000200,000000
 (1)                                    ;       PS = 200,       STACK POINTER = R4
 (1)                                    ;*************************************************************************
 (1)
 (1)  002266  104400                            SCOPE
 (1)  002270  004567  013070            TST11:  JSR     R5,     PUSHR   ;PUSH 4 WORDS ONTO R4 STACK, SET PRIORITY
 (1)  002274  100252  125252                    .WORD   100252,125252   ;SECOND OPERAND ON TOP
 (1)  002300  000425  052525                    .WORD   000425,052525   ;FIRST OPERAND ON BOTTOM
 (1)  002304  000217                            .WORD   217             ;PROCESSOR PRIORITY LEVEL
 (1)  002306  015634  000340                    .WORD   TRAPER,340      ;FIS TRAP VECTOR
 (1)  002312  012704  000510                    MOV     #STACK0,R4      ;SET UP STACK POINTER
 (1)
 (1)  002316  000240                            NOP
 (1)  002320  075004                            FADD    R4              ;FLOATING ADD ON THE R4 STACK
 (1)
 (1)  002322  004767  013070                    JSR     PC,     POPR    ;POP THE ANSWER
 (1)  002326  010467  176102                    MOV     R4,     $SP     ;SAVE "STACK POINTER"
 (1)  002332  122767  000200  176072            CMPB    #200,   $PSW    ;CHECK PS (EXCEPT T BIT)
 (1)  002340  001402                            BEQ     .+6             ;BRANCH IF OK
 (1)  002342  104000                            HLT                     ;PS NOT EQUAL TO 200
 (3)  002344  000051                            51                      ;THE ERROR NUMBER IS 51
 (1)
 (1)  002346  022767  000514  176060            CMP     #STACK4,$SP     ;CHECK THE STACK POINTER (R4)
 (1)  002354  001402                            BEQ     .+6             ;BRANCH IF OK
 (1)  002356  104000                            HLT                     ;STACK POINTER (R4) NOT EQUAL TO #STACK4
 (3)  002360  000052                            52                      ;THE ERROR NUMBER IS 52
 (1)
 (1)  002362  022767  000200  176046            CMP     #000200,ANS1    ;CHECK FIRST HALF OF ANSWER
 (1)  002370  001402                            BEQ     .+6             ;BRANCH IF OK
 (1)  002372  104002                            HLT+2                   ;ANS1 NOT EQUAL TO 000200
 (3)  002374  000053                            53                      ;THE ERROR NUMBER IS 53
 (1)
 (1)  002376  005767  176036                    TST     ANS2            ;CHECK SECOND HALF OF ANSWER
 (1)  002402  001402                            BEQ     .+6             ;BRANCH IF OK
 (1)  002404  104002                            HLT+2                   ;ANS2 NOT EQUAL TO 000000
 (3)  002406  000054                            54                      ;THE ERROR NUMBER IS 54
 (1)
 (1)  002410  122767  000011  175766    END11:  CMPB    #11,    $TESTN  ;CHECK THE TEST NUMBER
 (1)  002416  001402                            BEQ     .+6             ;BRANCH IF OK
 (1)  002420  104000                            HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
 (3)  002422  000055                            55                      ;THE ERROR NUMBER IS 55
 (1)
 (1)
```

```
7156
 (1)                                ;*****************************************************************************
 (1)                                ;TEST 12:       FADD (LSI-11 FLOATING ADD INSTRUCTION)
 (1)                                ;       100425,052525 + 000252,125252 = 100200,000000
 (1)                                ;       PS = 210,       STACK POINTER = SP
 (1)                                ;*****************************************************************************
 (1)
 (1)  002424  104400                            SCOPE
 (1)  002426  004567  012560                    TST12:  JSR     R5,     PUSHS   ;PUSH 4 WORDS ONTO STACK, SET PRIORITY
 (1)  002432  000252  125252                    .WORD   000252,125252   ;SECOND OPERAND ON TOP
 (1)  002436  100425  052525                    .WORD   100425,052525   ;FIRST OPERAND ON BOTTOM
 (1)  002442  000307                            .WORD   307             ;PROCESSOR PRIORITY LEVEL
 (1)  002444  015634  000340                    .WORD   TRAPER,340      ;FIS TRAP VECTOR
 (1)
 (1)  002450  000240                            NOP
 (1)  002452  075006                            FADD    SP              ;FLOATING ADD ON THE STACK
 (1)
 (1)  002454  004767  012572                    JSR     PC,     POPS    ;POP THE ANSWER
 (1)  002460  022706  000600                    CMP     #BEGIN, SP      ;CHECK THE STACK POINTER
 (1)  002464  001405                            BEQ     TSA12           ;BRANCH IF OK
 (1)  002466  012706  000600                    MOV     #BEGIN, SP      ;RESTORE STACK POINTER
 (1)  002472  104000                            HLT                     ;STACK POINTER FOULED UP
 (3)  002474  000056                            56                      ;THE ERROR NUMBER IS 56
 (1)  002476  000421                            BR      END12           ;SKIP REST OF TEST
 (1)
 (1)  002500  122767  000210  175724  TSA12:    CMPB    #210,   $PSW    ;CHECK PS (EXCEPT T BIT)
 (1)  002506  001402                            BEQ     .+6             ;BRANCH IF OK
 (1)  002510  104000                            HLT                     ;PS NOT EQUAL TO 210
 (3)  002512  000057                            57                      ;THE ERROR NUMBER IS 57
 (1)
 (1)  002514  022767  100200  175714            CMP     #100200,ANS1    ;CHECK FIRST HALF OF ANSWER
 (1)  002522  001402                            BEQ     .+6             ;BRANCH IF OK
 (1)  002524  104002                            HLT+2                   ;ANS1 NOT EQUAL TO 100200
 (3)  002526  000060                            60                      ;THE ERROR NUMBER IS 60
 (1)
 (1)  002530  005767  175704                    TST     ANS2            ;CHECK SECOND HALF OF ANSWER
 (1)  002534  001402                            BEQ     .+6             ;BRANCH IF OK
 (1)  002536  104002                            HLT+2                   ;ANS2 NOT EQUAL TO 000000
 (3)  002540  000061                            61                      ;THE ERROR NUMBER IS 61
 (1)
 (1)  002542  122767  000012  175634  END12:    CMPB    #12,    $TESTN  ;CHECK THE TEST NUMBER
 (1)  002550  001402                            BEQ     .+6             ;BRANCH IF OK
 (1)  002552  104000                            HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
 (3)  002554  000062                            62                      ;THE ERROR NUMBER IS 62
 (1)
 (1)
```

```
7157
 (1)                                ;*************************************************************************
 (1)                                ;TEST 13:       FADD (LSI-11 FLOATING ADD INSTRUCTION)
 (1)                                ;       077652,125252 + 077452,125252 = 077777,177777
 (1)                                ;       PS = 200,       STACK POINTER = SP
 (1)                                ;*************************************************************************
 (1)
 (1)  002556  104400                            SCOPE
 (1)  002560  004567  012426                    TST13:  JSR     R5,     PUSHS   ;PUSH 4 WORDS ONTO STACK, SET PRIORITY
 (1)  002564  077452  125252                    .WORD   077452,125252   ;SECOND OPERAND ON TOP
 (1)  002570  077652  125252                    .WORD   077652,125252   ;FIRST OPERAND ON BOTTOM
 (1)  002574  000257                            .WORD   257             ;PROCESSOR PRIORITY LEVEL
 (1)  002576  015634  000340                    .WORD   TRAPER,340      ;FIS TRAP VECTOR
 (1)
 (1)  002602  000240                            NOP
 (1)  002604  075006                            FADD    SP              ;FLOATING ADD ON THE STACK
 (1)
 (1)  002606  004767  012440                    JSR     PC,     POPS    ;POP THE ANSWER
 (1)  002612  022706  000600                    CMP     #BEGIN, SP      ;CHECK THE STACK POINTER
 (1)  002616  001405                            BEQ     TSA13           ;BRANCH IF OK
 (1)  002620  012706  000600                    MOV     #BEGIN, SP      ;RESTORE STACK POINTER
 (1)  002624  104000                            HLT                     ;STACK POINTER FOULED UP
 (3)  002626  000063                            63                      ;THE ERROR NUMBER IS 63
 (1)  002630  000422                            BR      END13           ;SKIP REST OF TEST
 (1)
 (1)  002632  122767  000200  175572  TSA13:    CMPB    #200,   $PSW    ;CHECK PS (EXCEPT T BIT)
 (1)  002640  001402                            BEQ     .+6             ;BRANCH IF OK
 (1)  002642  104000                            HLT                     ;PS NOT EQUAL TO 200
 (3)  002644  000064                            64                      ;THE ERROR NUMBER IS 64
 (1)
 (1)  002646  022767  077777  175562            CMP     #077777,ANS1    ;CHECK FIRST HALF OF ANSWER
 (1)  002654  001402                            BEQ     .+6             ;BRANCH IF OK
 (1)  002656  104002                            HLT+2                   ;ANS1 NOT EQUAL TO 077777
 (3)  002660  000065                            65                      ;THE ERROR NUMBER IS 65
 (1)
 (1)  002662  022767  177777  175550            CMP     #177777,ANS2    ;CHECK SECOND HALF OF ANSWER
 (1)  002670  001402                            BEQ     .+6             ;BRANCH IF OK
 (1)  002672  104002                            HLT+2                   ;ANS2 NOT EQUAL TO 177777
 (3)  002674  000066                            66                      ;THE ERROR NUMBER IS 66
 (1)
 (1)  002676  122767  000013  175500  END13:    CMPB    #13,    $TESTN  ;CHECK THE TEST NUMBER
 (1)  002704  001402                            BEQ     .+6             ;BRANCH IF OK
 (1)  002706  104000                            HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
 (3)  002710  000067                            67                      ;THE ERROR NUMBER IS 67
 (1)
 (1)
```

```
7158
 (1)                                ;*************************************************************************
 (1)                                ;TEST 14:       FADD (LSI-11 FLOATING ADD INSTRUCTION)
 (1)                                ;       177452,125252 + 177652,125252 = 177777,177777
 (1)                                ;       PS = 210,       STACK POINTER = R4
 (1)                                ;*************************************************************************
 (1)
 (1)  002712  104400                          SCOPE
 (1)  002714  004567  012444          TST14:  JSR     R5,     PUSHR   ;PUSH 4 WORDS ONTO R4 STACK, SET PRIORITY
 (1)  002720  177652  125252                  .WORD   177652,125252   ;SECOND OPERAND ON TOP
 (1)  002724  177452  125252                  .WORD   177452,125252   ;FIRST OPERAND ON BOTTOM
 (1)  002730  000357                          .WORD   357             ;PROCESSOR PRIORITY LEVEL
 (1)  002732  015634  000340                  .WORD   TRAPER,340      ;FIS TRAP VECTOR
 (1)  002736  012704  000510                  MOV     #STACK0,R4      ;SET UP STACK POINTER
 (1)
 (1)  002742  000240                          NOP
 (1)  002744  075004                          FADD    R4              ;FLOATING ADD ON THE R4 STACK
 (1)
 (1)  002746  004767  012444                  JSR     PC,     POPR    ;POP THE ANSWER
 (1)  002752  010467  175456                  MOV     R4,     $SP     ;SAVE "STACK POINTER"
 (1)  002756  122767  000210  175446          CMPB    #210,   $PSW    ;CHECK PS (EXCEPT T BIT)
 (1)  002764  001402                          BEQ     .+6             ;BRANCH IF OK
 (1)  002766  104000                          HLT                     ;PS NOT EQUAL TO 210
 (3)  002770  000070                          70                      ;THE ERROR NUMBER IS 70
 (1)
 (1)  002772  022767  000514  175434          CMP     #STACK4,$SP     ;CHECK THE STACK POINTER (R4)
 (1)  003000  001402                          BEQ     .+6             ;BRANCH IF OK
 (1)  003002  104000                          HLT                     ;STACK POINTER (R4) NOT EQUAL TO #STACK4
 (3)  003004  000071                          71                      ;THE ERROR NUMBER IS 71
 (1)
 (1)  003006  022767  177777  175422          CMP     #177777,ANS1    ;CHECK FIRST HALF OF ANSWER
 (1)  003014  001402                          BEQ     .+6             ;BRANCH IF OK
 (1)  003016  104002                          HLT+2                   ;ANS1 NOT EQUAL TO 177777
 (3)  003020  000072                          72                      ;THE ERROR NUMBER IS 72
 (1)
 (1)  003022  022767  177777  175410          CMP     #177777,ANS2    ;CHECK SECOND HALF OF ANSWER
 (1)  003030  001402                          BEQ     .+6             ;BRANCH IF OK
 (1)  003032  104002                          HLT+2                   ;ANS2 NOT EQUAL TO 177777
 (3)  003034  000073                          73                      ;THE ERROR NUMBER IS 73
 (1)
 (1)  003036  122767  000014  175340  END14:  CMPB    #14,    $TESTN  ;CHECK THE TEST NUMBER
 (1)  003044  001402                          BEQ     .+6             ;BRANCH IF OK
 (1)  003046  104000                          HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
 (3)  003050  000074                          74                      ;THE ERROR NUMBER IS 74
 (1)
 (1)
```

```
7159
 (1)                            ;*****************************************************************************
 (1)                            ;TEST 15:       FADD (LSI-11 FLOATING ADD INSTRUCTION)
 (1)                            ;       004000,105004 + 104000,104000 = 000401,000000
 (1)                            ;       PS = 000,       STACK POINTER = PC
 (1)                            ;*****************************************************************************
 (1)
 (1)  003052  104400                           SCOPE
 (1)  003054  004567  012456    TST15:  JSR     R5,     PUSH7   ;PUSH 4 WORDS ONTO STACK, SET PRIORITY
 (1)  003060  003104                    .WORD   STK15           ;TOP OF STACK
 (1)  003062  104000  104000            .WORD   104000,104000   ;SECOND OPERAND ON TOP
 (1)  003066  004000  105004            .WORD   004000,105004   ;FIRST OPERAND ON BOTTOM
 (1)  003072  000144                    .WORD   144             ;PROCESSOR PRIORITY LEVEL
 (1)  003074  015634  000340            .WORD   TRAPER,340      ;FIS TRAP VECTOR
 (1)
 (1)  003100  000240                    NOP
 (1)  003102  075007                    FADD    PC              ;FLOATING ADD ON FOLLOWING 4 WORDS
 (1)  003104  104000    STK15:  104000          ;SHOULD CONTAIN 104000
 (1)  003106  104000            104000          ;SHOULD CONTAIN 104000
 (1)  003110  004000            004000          ;BEFORE FADD, 004000; AFTER, 000401
 (1)  003112  105004            105004          ;BEFORE FADD, 105004; AFTER, 000000
 (1)
 (1)  003114  004767  012446            JSR     PC,     POP7    ;POP THE ANSWER
 (1)  003120  105767  175306            TSTB    $PSW            ;CHECK PS (EXCEPT T BIT)
 (1)  003124  001402                    BEQ     .+6             ;BRANCH IF OK
 (1)  003126  104000                    HLT                     ;PS NOT EQUAL TO 000
 (3)  003130  000075                    75                      ;THE ERROR NUMBER IS 75
 (1)
 (1)  003132  022767  104000  175276    CMP     #104000,ANS1    ;CHECK FIRST HALF OF INPUT DATA (STK15)
 (1)  003140  001402                    BEQ     .+6             ;BRANCH IF OK
 (1)  003142  104002                    HLT+2                   ;ANS1 NOT EQUAL TO 104000
 (3)  003144  000076                    76                      ;THE ERROR NUMBER IS 76
 (1)
 (1)  003146  022767  104000  175264    CMP     #104000,ANS2    ;CHECK SECOND HALF OF INPUT DATA (STK15+2)
 (1)  003154  001402                    BEQ     .+6             ;BRANCH IF OK
 (1)  003156  104002                    HLT+2                   ;ANS2 NOT EQUAL TO 104000
 (3)  003160  000077                    77                      ;THE ERROR NUMBER IS 77
 (1)
 (1)  003162  022767  000401  175252    CMP     #000401,ANS3    ;CHECK FIRST HALF OF ANSWER
 (1)  003170  001402                    BEQ     .+6             ;BRANCH IF OK
 (1)  003172  104004                    HLT+4                   ;ANS3 NOT EQUAL TO 000401
 (3)  003174  000100                    100                     ;THE ERROR NUMBER IS 100
 (1)
 (1)  003176  005767  175242            TST     ANS4            ;CHECK SECOND HALF OF ANSWER
 (1)  003202  001402                    BEQ     .+6             ;BRANCH IF OK
 (1)  003204  104004                    HLT+4                   ;ANS4 NOT EQUAL TO 000000
 (3)  003206  000101                    101                     ;THE ERROR NUMBER IS 101
 (1)
 (1)  003210  122767  000015  175166  END15:  CMPB    #15,    $TESTN  ;CHECK THE TEST NUMBER
 (1)  003216  001402                    BEQ     .+6             ;BRANCH IF OK
 (1)  003220  104000                    HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
 (3)  003222  000102                    102                     ;THE ERROR NUMBER IS 102
 (1)
 (1)
7163
 (1)                            ;*****************************************************************************
```

```
(1)                                     ;TEST 16:       FADD (LSI-11 FLOATING ADD INSTRUCTION)
(1)                                     ;       100200,000000 + 000377,177777 ==> UNDERFLOW
(1)                                     ;       PS(ON STACK) = 012,     STACK POINTER = R3
(1)                                     ;*********************************************************************
(1)
(1)  003224  104400                             SCOPE
(1)  003226  004567  012132             TST16:  JSR     R5,     PUSHR   ;PUSH 4 WORDS ONTO R3 STACK, SET PRIORITY
(1)  003232  000377  177777                     .WORD   000377,177777   ;SECOND OPERAND ON TOP
(1)  003236  100200  000000                     .WORD   100200,000000   ;FIRST OPERAND ON BOTTOM
(1)  003242  000157                             .WORD   157             ;PROCESSOR PRIORITY LEVEL
(1)  003244  003276  000000                     .WORD   ISR16,  000     ;FIS TRAP VECTOR
(1)  003250  012703  000510                     MOV     #STACK0,R3      ;SET UP R3 AS STACK POINTER
(1)
(1)  003254  000240                             NOP
(1)  003256  075003                             FADD    R3              ;FLOATING ADD ON THE R3 STACK
(1)
(1)  003260  004767  012132             RTA16:  JSR     PC,     POPR    ;POP THE "ANSWER"
(1)  003264  010367  175144                     MOV     R3,     $SP     ;SAVE STACK POINTER (R3)
(1)  003270  104002                             HLT+2                   ;FIS TRAP DIDN'T OCCURE!
(3)  003272  000103                             103                     ;THE ERROR NUMBER IS 103
(1)  003274  000462                             BR      END16
(1)
(1)  003276  004767  012144             ISR16:  JSR     PC,     POPER   ;POP ALL DATA OFF THE STACKS
(1)  003302  010367  175126                     MOV     R3,     $SP     ;SAVE STACK POINTER (R3)
(1)  003306  105767  175120                     TSTB    $PSW            ;CHECK PS AFTER FIS TRAP
(1)  003312  001402                             BEQ     .+6             ;BRANCH IF OK
(1)  003314  104000                             HLT                     ;PS AFTER FIS TRAP NOT EQUAL TO 000
(3)  003316  000104                             104                     ;THE ERROR NUMBER IS 104
(1)
(1)  003320  022767  000510  175106             CMP     #STACK0,$SP     ;CHECK THE STACK POINTER (R3)
(1)  003326  001402                             BEQ     .+6             ;BRANCH IF OK
(1)  003330  104000                             HLT                     ;STACK POINTER (R3) NOT EQUAL TO #STACK0
(3)  003332  000105                             105                     ;THE ERROR NUMBER IS 105
(1)
(1)  003334  022767  003260  175074             CMP     #RTA16, ANS1    ;CHECK FIS TRAP RETURN ADDRESS
(1)  003342  001402                             BEQ     .+6             ;BRANCH IF OK
(1)  003344  104001                             HLT+1                   ;FIS TRAP AT WRONG ADDRESS
(3)  003346  000106                             106                     ;THE ERROR NUMBER IS 106
(1)
(1)  003350  022767  000012  175062             CMP     #012,   ANS2    ;CHECK PS BEFORE FIS TRAP
(1)  003356  001402                             BEQ     .+6             ;BRANCH IF OK
(1)  003360  104002                             HLT+2                   ;PS AT FIS TRAP TIME NOT 012
(3)  003362  000107                             107                     ;THE ERROR NUMBER IS 107
(1)
(1)  003364  022767  000377  175050             CMP     #000377,ANS3    ;CHECK DATA FROM THE STACK
(1)  003372  001402                             BEQ     .+6             ;BRANCH IF OK
(1)  003374  104004                             HLT+4                   ;DATA ON STACK (000377) CHANGED
(3)  003376  000110                             110                     ;THE ERROR NUMBER IS 110
(1)
(1)  003400  022767  177777  175036             CMP     #177777,ANS4    ;CHECK DATA FROM STACK
(1)  003406  001402                             BEQ     .+6             ;BRANCH IF OK
(1)  003410  104004                             HLT+4                   ;DATA ON STACK (177777) CHANGED
(3)  003412  000111                             111                     ;THE ERROR NUMBER IS 111
(1)
(1)  003414  022767  100200  175024             CMP     #100200,ANS5    ;CHECK DATA FROM STACK
(1)  003422  001402                             BEQ     .+6             ;BRANCH IF OK
```

```
    (1)  003424  104006                          HLT+6                   ;DATA ON STACK (100200) CHANGED
    (3)  003426  000112                          112                     ;THE ERROR NUMBER IS 112
    (1)
    (1)  003430  005767  175014                  TST     ANS6            ;CHECK DATA FROM STACK
    (1)  003434  001402                          BEQ     .+6             ;BRANCH IF OK
    (1)  003436  104006                          HLT+6                   ;DATA ON STACK (000000) CHANGED
    (3)  003440  000113                          113                     ;THE ERROR NUMBER IS 113
    (1)
    (1)  003442  122767  000016  174734  END16:  CMPB    #16,    $TESTN  ;CHECK THE TEST NUMBER
    (1)  003450  001402                          BEQ     .+6             ;BRANCH IF OK
    (1)  003452  104000                          HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
    (3)  003454  000114                          114                     ;THE ERROR NUMBER IS 114
    (1)
    (1)
```

```
7164
 (1)                            ;*************************************************************************
 (1)                            ;TEST 17:       FADD (LSI-11 FLOATING ADD INSTRUCTION)
 (1)                            ;       000200,000000 + 100377,177777 ==> UNDERFLOW
 (1)                            ;       PS(ON STACK) = 212,     STACK POINTER = SP
 (1)                            ;*************************************************************************
 (1)
 (1)  003456  104400                    SCOPE
 (1)  003460  004567  011526    TST17:  JSR     R5,     PUSHS   ;PUSH 4 WORDS ONTO STACK, SET PRIORITY
 (1)  003464  100377  177777            .WORD   100377,177777   ;SECOND OPERAND ON TOP
 (1)  003470  000200  000000            .WORD   000200,000000   ;FIRST OPERAND ON BOTTOM
 (1)  003474  000257                    .WORD   257             ;PROCESSOR PRIORITY LEVEL
 (1)  003476  003524  000340            .WORD   ISR17,  340     ;FIS TRAP VECTOR
 (1)
 (1)  003502  000240                    NOP
 (1)  003504  075006                    FADD    SP              ;FLOATING ADD ON THE STACK
 (1)
 (1)  003506  004767  011540    RTA17:  JSR     PC,     POPS    ;POP THE "ANSWER"
 (1)  003512  104002                    HLT+2                   ;FIS TRAP DIDN'T OCCURE!
 (3)  003514  000115                    115                     ;THE ERROR NUMBER IS 115
 (1)  003516  012706  000600            MOV     #BEGIN, SP      ;RESTORE THE STACK POINTER
 (1)  003522  000463                    BR      END17
 (1)
 (1)  003524  004767  011554    ISR17:  JSR     PC,     POPES   ;POP ALL DATA OFF THE STACK
 (1)  003530  022706  000600            CMP     #BEGIN, SP      ;CHECK THE STACK POINTER
 (1)  003534  001405                    BEQ     ISA17           ;BRANCH IF OK
 (1)  003536  012706  000600            MOV     #BEGIN, SP      ;RESTORE THE STACK POINTER
 (1)  003542  104000                    HLT                     ;STACK POINTER FOULED UP
 (3)  003544  000116                    116                     ;THE ERROR NUMBER IS 116
 (1)  003546  000451                    BR      END17           ;SKIP REST OF TEST
 (1)
 (1)  003550  122767  000340  174654  ISA17:  CMPB    #340,   $PSW    ;CHECK PS AFTER FIS TRAP
 (1)  003556  001402                    BEQ     .+6             ;BRANCH IF OK
 (1)  003560  104000                    HLT                     ;PS AFTER FIS TRAP NOT EQUAL TO 340
 (3)  003562  000117                    117                     ;THE ERROR NUMBER IS 117
 (1)
 (1)  003564  022767  003506  174644            CMP     #RTA17, ANS1    ;CHECK FIS TRAP RETURN ADDRESS
 (1)  003572  001402                            BEQ     .+6             ;BRANCH IF OK
 (1)  003574  104001                            HLT+1                   ;FIS TRAP AT WRONG ADDRESS
 (3)  003576  000120                            120                     ;THE ERROR NUMBER IS 120
 (1)
 (1)  003600  022767  000212  174632            CMP     #212,   ANS2    ;CHECK PS BEFORE FIS TRAP
 (1)  003606  001402                            BEQ     .+6             ;BRANCH IF OK
 (1)  003610  104002                            HLT+2                   ;PS AT FIS TRAP TIME NOT 212
 (3)  003612  000121                            121                     ;THE ERROR NUMBER IS 121
 (1)
 (1)  003614  022767  100377  174620            CMP     #100377,ANS3    ;CHECK DATA FROM THE STACK
 (1)  003622  001402                            BEQ     .+6             ;BRANCH IF OK
 (1)  003624  104004                            HLT+4                   ;DATA ON STACK (100377) CHANGED
 (3)  003626  000122                            122                     ;THE ERROR NUMBER IS 122
 (1)
 (1)  003630  022767  177777  174606            CMP     #177777,ANS4    ;CHECK DATA FROM STACK
 (1)  003636  001402                            BEQ     .+6             ;BRANCH IF OK
 (1)  003640  104004                            HLT+4                   ;DATA ON STACK (177777) CHANGED
 (3)  003642  000123                            123                     ;THE ERROR NUMBER IS 123
 (1)
```

```
    (1)  003644  022767  000200  174574          CMP     #000200,ANS5    ;CHECK DATA FROM STACK
    (1)  003652  001402                          BEQ     .+6             ;BRANCH IF OK
    (1)  003654  104006                          HLT+6                   ;DATA ON STACK (000200) CHANGED
    (3)  003656  000124                          124                     ;THE ERROR NUMBER IS 124
    (1)
    (1)  003660  005767  174564                  TST     ANS6            ;CHECK DATA FROM STACK
    (1)  003664  001402                          BEQ     .+6             ;BRANCH IF OK
    (1)  003666  104006                          HLT+6                   ;DATA ON STACK (000000) CHANGED
    (3)  003670  000125                          125                     ;THE ERROR NUMBER IS 125
    (1)
    (1)  003672  122767  000017  174504  END17:  CMPB    #17,    $TESTN  ;CHECK THE TEST NUMBER
    (1)  003700  001402                          BEQ     .+6             ;BRANCH IF OK
    (1)  003702  104000                          HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
    (3)  003704  000126                          126                     ;THE ERROR NUMBER IS 126
    (1)
    (1)
```

```
 7168
  (1)                                ;*************************************************************************
  (1)                                ;TEST 20:       FADD (LSI-11 FLOATING ADD INSTRUCTION)
  (1)                                ;       177452,125253 + 177652,125252 ==> OVERFLOW
  (1)                                ;       PS(ON STACK) = 002,     STACK POINTER = R1
  (1)                                ;*************************************************************************
  (1)
  (1)  003706  104400                           SCOPE
  (1)  003710  004567  011450        TST20:     JSR     R5,     PUSHR   ;PUSH 4 WORDS ONTO R1 STACK, SET PRIORITY
  (1)  003714  177652  125252                   .WORD   177652,125252   ;SECOND OPERAND ON TOP
  (1)  003720  177452  125253                   .WORD   177452,125253   ;FIRST OPERAND ON BOTTOM
  (1)  003724  000105                           .WORD   105             ;PROCESSOR PRIORITY LEVEL
  (1)  003726  003760  000252                   .WORD   ISR20, 252      ;FIS TRAP VECTOR
  (1)  003732  012701  000510                   MOV     #STACK0,R1      ;SET UP R1 AS STACK POINTER
  (1)
  (1)  003736  000240                           NOP
  (1)  003740  075001                           FADD    R1              ;FLOATING ADD ON THE R1 STACK
  (1)
  (1)  003742  004767  011450        RTA20:     JSR     PC,     POPR    ;POP THE "ANSWER"
  (1)  003746  010167  174462                   MOV     R1,     $SP     ;SAVE STACK POINTER (R1)
  (1)  003752  104002                           HLT+2                   ;FIS TRAP DIDN'T OCCURE!
  (3)  003754  000127                           127                     ;THE ERROR NUMBER IS 127
  (1)  003756  000464                           BR      END20
  (1)
  (1)  003760  004767  011462        ISR20:     JSR     PC,     POPER   ;POP ALL DATA OFF THE STACKS
  (1)  003764  010167  174444                   MOV     R1,     $SP     ;SAVE STACK POINTER (R1)
  (1)  003770  122767  000252  174434           CMPB    #252,   $PSW    ;CHECK PS AFTER FIS TRAP
  (1)  003776  001402                           BEQ     .+6             ;BRANCH IF OK
  (1)  004000  104000                           HLT                     ;PS AFTER FIS TRAP NOT EQUAL TO 252
  (3)  004002  000130                           130                     ;THE ERROR NUMBER IS 130
  (1)
  (1)  004004  022767  000510  174422           CMP     #STACK0,$SP     ;CHECK THE STACK POINTER (R1)
  (1)  004012  001402                           BEQ     .+6             ;BRANCH IF OK
  (1)  004014  104000                           HLT                     ;STACK POINTER (R1) NOT EQUAL TO #STACK0
  (3)  004016  000131                           131                     ;THE ERROR NUMBER IS 131
  (1)
  (1)  004020  022767  003742  174410           CMP     #RTA20, ANS1    ;CHECK FIS TRAP RETURN ADDRESS
  (1)  004026  001402                           BEQ     .+6             ;BRANCH IF OK
  (1)  004030  104001                           HLT+1                   ;FIS TRAP AT WRONG ADDRESS
  (3)  004032  000132                           132                     ;THE ERROR NUMBER IS 132
  (1)
  (1)  004034  022767  000002  174376           CMP     #002,   ANS2    ;CHECK PS BEFORE FIS TRAP
  (1)  004042  001402                           BEQ     .+6             ;BRANCH IF OK
  (1)  004044  104002                           HLT+2                   ;PS AT FIS TRAP TIME NOT 002
  (3)  004046  000133                           133                     ;THE ERROR NUMBER IS 133
  (1)
  (1)  004050  022767  177652  174364           CMP     #177652,ANS3    ;CHECK DATA FROM THE STACK
  (1)  004056  001402                           BEQ     .+6             ;BRANCH IF OK
  (1)  004060  104004                           HLT+4                   ;DATA ON STACK (177652) CHANGED
  (3)  004062  000134                           134                     ;THE ERROR NUMBER IS 134
  (1)
  (1)  004064  022767  125252  174352           CMP     #125252,ANS4    ;CHECK DATA FROM STACK
  (1)  004072  001402                           BEQ     .+6             ;BRANCH IF OK
  (1)  004074  104004                           HLT+4                   ;DATA ON STACK (125252) CHANGED
  (3)  004076  000135                           135                     ;THE ERROR NUMBER IS 135
  (1)
```

```
    (1)  004100  022767  177452  174340          CMP     #177452,ANS5    ;CHECK DATA FROM STACK
    (1)  004106  001402                          BEQ     .+6             ;BRANCH IF OK
    (1)  004110  104006                          HLT+6                   ;DATA ON STACK (177452) CHANGED
    (3)  004112  000136                          136                     ;THE ERROR NUMBER IS 136
    (1)
    (1)  004114  022767  125253  174326          CMP     #125253,ANS6    ;CHECK DATA FROM STACK
    (1)  004122  001402                          BEQ     .+6             ;BRANCH IF OK
    (1)  004124  104006                          HLT+6                   ;DATA ON STACK (125253) CHANGED
    (3)  004126  000137                          137                     ;THE ERROR NUMBER IS 137
    (1)
    (1)  004130  122767  000020  174246  END20:  CMPB    #20,    STESTN  ;CHECK THE TEST NUMBER
    (1)  004136  001402                          BEQ     .+6             ;BRANCH IF OK
    (1)  004140  104000                          HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
    (3)  004142  000140                          140                     ;THE ERROR NUMBER IS 140
    (1)
    (1)
```

```
7169
 (1)                                ;*************************************************************************
 (1)                                ;TEST 21:       FADD (LSI-11 FLOATING ADD INSTRUCTION)
 (1)                                ;       077652,125253 + 077452,125252 ==> OVERFLOW
 (1)                                ;       PS(ON STACK) = 002,     STACK POINTER = SP
 (1)                                ;*************************************************************************
 (1)
 (1)  004144  104400                            SCOPE
 (1)  004146  004567  011040        TST21:      JSR     R5,     PUSHS   ;PUSH 4 WORDS ONTO STACK, SET PRIORITY
 (1)  004152  077452  125252                    .WORD   077452,125252   ;SECOND OPERAND ON TOP
 (1)  004156  077652  125253                    .WORD   077652,125253   ;FIRST OPERAND ON BOTTOM
 (1)  004162  000003                            .WORD   003             ;PROCESSOR PRIORITY LEVEL
 (1)  004164  004212  000344                    .WORD   ISR21,  344     ;FIS TRAP VECTOR
 (1)
 (1)  004170  000240                            NOP
 (1)  004172  075006                            FADD    SP              ;FLOATING ADD ON THE STACK
 (1)
 (1)  004174  004767  011052        RTA21:      JSR     PC,     POPS    ;POP THE "ANSWER"
 (1)  004200  104002                            HLT+2                   ;FIS TRAP DIDN'T OCCURE!
 (3)  004202  000141                            141                     ;THE ERROR NUMBER IS 141
 (1)  004204  012706  000600                    MOV     #BEGIN, SP      ;RESTORE THE STACK POINTER
 (1)  004210  000464                            BR      END21
 (1)
 (1)  004212  004767  011066        ISR21:      JSR     PC,     POPES   ;POP ALL DATA OFF THE STACK
 (1)  004216  022706  000600                    CMP     #BEGIN, SP      ;CHECK THE STACK POINTER
 (1)  004222  001405                            BEQ     ISA21           ;BRANCH IF OK
 (1)  004224  012706  000600                    MOV     #BEGIN, SP      ;RESTORE THE STACK POINTER
 (1)  004230  104000                            HLT                     ;STACK POINTER FOULED UP
 (3)  004232  000142                            142                     ;THE ERROR NUMBER IS 142
 (1)  004234  000452                            BR      END21           ;SKIP REST OF TEST
 (1)
 (1)  004236  122767  000344  174166  ISA21:     CMPB    #344,   SPSW    ;CHECK PS AFTER FIS TRAP
 (1)  004244  001402                            BEQ     .+6             ;BRANCH IF OK
 (1)  004246  104000                            HLT                     ;PS AFTER FIS TRAP NOT EQUAL TO 344
 (3)  004250  000143                            143                     ;THE ERROR NUMBER IS 143
 (1)
 (1)  004252  022767  004174  174156            CMP     #RTA21, ANS1    ;CHECK FIS TRAP RETURN ADDRESS
 (1)  004260  001402                            BEQ     .+6             ;BRANCH IF OK
 (1)  004262  104001                            HLT+1                   ;FIS TRAP AT WRONG ADDRESS
 (3)  004264  000144                            144                     ;THE ERROR NUMBER IS 144
 (1)
 (1)  004266  022767  000002  174144            CMP     #002,   ANS2    ;CHECK PS BEFORE FIS TRAP
 (1)  004274  001402                            BEQ     .+6             ;BRANCH IF OK
 (1)  004276  104002                            HLT+2                   ;PS AT FIS TRAP TIME NOT 002
 (3)  004300  000145                            145                     ;THE ERROR NUMBER IS 145
 (1)
 (1)  004302  022767  077452  174132            CMP     #077452,ANS3    ;CHECK DATA FROM THE STACK
 (1)  004310  001402                            BEQ     .+6             ;BRANCH IF OK
 (1)  004312  104004                            HLT+4                   ;DATA ON STACK (077452) CHANGED
 (3)  004314  000146                            146                     ;THE ERROR NUMBER IS 146
 (1)
 (1)  004316  022767  125252  174120            CMP     #125252,ANS4    ;CHECK DATA FROM STACK
 (1)  004324  001402                            BEQ     .+6             ;BRANCH IF OK
 (1)  004326  104004                            HLT+4                   ;DATA ON STACK (125252) CHANGED
 (3)  004330  000147                            147                     ;THE ERROR NUMBER IS 147
 (1)
```

```
(1)  004332  022767  077652  174106          CMP     #077652,ANS5    ;CHECK DATA FROM STACK
(1)  004340  001402                          BEQ     .+6             ;BRANCH IF OK
(1)  004342  104006                          HLT+6                   ;DATA ON STACK (077652) CHANGED
(3)  004344  000150                          150                     ;THE ERROR NUMBER IS 150
(1)
(1)  004346  022767  125253  174074          CMP     #125253,ANS6    ;CHECK DATA FROM STACK
(1)  004354  001402                          BEQ     .+6             ;BRANCH IF OK
(1)  004356  104006                          HLT+6                   ;DATA ON STACK (125253) CHANGED
(3)  004360  000151                          151                     ;THE ERROR NUMBER IS 151
(1)
(1)  004362  122767  000021  174014  END21:  CMPB    #21,    STESTN  ;CHECK THE TEST NUMBER
(1)  004370  001402                          BEQ     .+6             ;BRANCH IF OK
(1)  004372  104000                          HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
(3)  004374  000152                          152                     ;THE ERROR NUMBER IS 152
(1)
(1)
```

```
7173
 (1)                                ;*************************************************************************
 (1)                                ;TEST 22:       FSUB (LSI-11 FLOATING SUBTRACT INSTRUCTION)
 (1)                                ;       177520,017552 - 135352,051107 = 177520,017552
 (1)                                ;       PS = 010,       STACK POINTER = R1
 (1)                                ;*************************************************************************
 (1)
 (1) 004376  104400                            SCOPE
 (1) 004400  004567  010760         TST22:     JSR     R5,     PUSHR   ;PUSH 4 WORDS ONTO R1 STACK, SET PRIORITY
 (1) 004404  135352  051107                    .WORD   135352,051107   ;SECOND OPERAND ON TOP
 (1) 004410  177520  017552                    .WORD   177520,017552   ;FIRST OPERAND ON BOTTOM
 (1) 004414  000040                            .WORD   040             ;PROCESSOR PRIORITY LEVEL
 (1) 004416  015634  000340                    .WORD   TRAPER, 340     ;FIS TRAP VECTOR
 (1) 004422  012701  000510                    MOV     #STACK0,R1      ;SET UP STACK POINTER
 (1)
 (1) 004426  000240                            NOP
 (1) 004430  075011                            FSUB    R1              ;FLOATING SUBTRACT ON THE R1 STACK
 (1)
 (1) 004432  004767  010760                    JSR     PC,     POPR    ;POP THE ANSWER
 (1) 004436  010167  173772                    MOV     R1,     $SP     ;SAVE "STACK POINTER"
 (1) 004442  122767  000010  173762            CMPB    #010,   $PSW    ;CHECK PS (EXCEPT T BIT)
 (1) 004450  001402                            BEQ     .+6             ;BRANCH IF OK
 (1) 004452  104000                            HLT                     ;PS NOT EQUAL TO 010
 (3) 004454  000153                            153                     ;THE ERROR NUMBER IS 153
 (1)
 (1) 004456  022767  000514  173750            CMP     #STACK4,$SP     ;CHECK THE STACK POINTER (R1)
 (1) 004464  001402                            BEQ     .+6             ;BRANCH IF OK
 (1) 004466  104000                            HLT                     ;STACK POINTER (R1) NOT EQUAL TO #STACK4
 (3) 004470  000154                            154                     ;THE ERROR NUMBER IS 154
 (1)
 (1) 004472  022767  177520  173736            CMP     #177520,ANS1    ;CHECK FIRST HALF OF ANSWER
 (1) 004500  001402                            BEQ     .+6             ;BRANCH IF OK
 (1) 004502  104002                            HLT+2                   ;ANS1 NOT EQUAL TO 177520
 (3) 004504  000155                            155                     ;THE ERROR NUMBER IS 155
 (1)
 (1) 004506  022767  017552  173724            CMP     #017552,ANS2    ;CHECK SECOND HALF OF ANSWER
 (1) 004514  001402                            BEQ     .+6             ;BRANCH IF OK
 (1) 004516  104002                            HLT+2                   ;ANS2 NOT EQUAL TO 017552
 (3) 004520  000156                            156                     ;THE ERROR NUMBER IS 156
 (1)
 (1) 004522  122767  000022  173654  END22:     CMPB    #22,    $TESTN  ;CHECK THE TEST NUMBER
 (1) 004530  001402                            BEQ     .+6             ;BRANCH IF OK
 (1) 004532  104000                            HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
 (3) 004534  000157                            157                     ;THE ERROR NUMBER IS 157
 (1)
 (1)
```

```
 7174
  (1)
  (1)                                   ;*****************************************************************************
  (1)                                   ;TEST 23:       FSUB (LSI-11 FLOATING SUBTRACT INSTRUCTION)
  (1)                                   ;       125252,125252 - 125252,125253 = 017400,000000
  (1)                                   ;       PS = 000,       STACK POINTER = R0
  (1)                                   ;*****************************************************************************
  (1)
  (1)  004536  104400                           SCOPE
  (1)  004540  004567  010620           TST23:  JSR     R5,     PUSHR   ;PUSH 4 WORDS ONTO R0 STACK, SET PRIORITY
  (1)  004544  125252  125253                   .WORD   125252,125253   ;SECOND OPERAND ON TOP
  (1)  004550  125252  125252                   .WORD   125252,125252   ;FIRST OPERAND ON BOTTOM
  (1)  004554  000047                           .WORD   047             ;PROCESSOR PRIORITY LEVEL
  (1)  004556  015634  000340                   .WORD   TRAPER, 340     ;FIS TRAP VECTOR
  (1)  004562  012700  000510                   MOV     #STACK0,R0      ;SET UP STACK POINTER
  (1)
  (1)  004566  000240                           NOP
  (1)  004570  075010                           FSUB    R0              ;FLOATING SUBTRACT ON THE R0 STACK
  (1)
  (1)  004572  004767  010620                   JSR     PC,     POPR    ;POP THE ANSWER
  (1)  004576  010067  173632                   MOV     R0,     $SP     ;SAVE "STACK POINTER"
  (1)  004602  105767  173624                   TSTB    $PSW            ;CHECK PS (EXCEPT T BIT)
  (1)  004606  001402                           BEQ     .+6             ;BRANCH IF OK
  (1)  004610  104000                           HLT                     ;PS NOT EQUAL TO 000
  (3)  004612  000160                           160                     ;THE ERROR NUMBER IS 160
  (1)
  (1)  004614  022767  000514  173612           CMP     #STACK4,$SP     ;CHECK THE STACK POINTER (R0)
  (1)  004622  001402                           BEQ     .+6             ;BRANCH IF OK
  (1)  004624  104000                           HLT                     ;STACK POINTER (R0) NOT EQUAL TO #STACK4
  (3)  004626  000161                           161                     ;THE ERROR NUMBER IS 161
  (1)
  (1)  004630  022767  017400  173600           CMP     #017400,ANS1    ;CHECK FIRST HALF OF ANSWER
  (1)  004636  001402                           BEQ     .+6             ;BRANCH IF OK
  (1)  004640  104002                           HLT+2                   ;ANS1 NOT EQUAL TO 017400
  (3)  004642  000162                           162                     ;THE ERROR NUMBER IS 162
  (1)
  (1)  004644  005767  173570                   TST     ANS2            ;CHECK SECOND HALF OF ANSWER
  (1)  004650  001402                           BEQ     .+6             ;BRANCH IF OK
  (1)  004652  104002                           HLT+2                   ;ANS2 NOT EQUAL TO 000000
  (3)  004654  000163                           163                     ;THE ERROR NUMBER IS 163
  (1)
  (1)  004656  122767  000023  173520   END23:  CMPB    #23,    $TESTN  ;CHECK THE TEST NUMBER
  (1)  004664  001402                           BEQ     .+6             ;BRANCH IF OK
  (1)  004666  104000                           HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
  (3)  004670  000164                           164                     ;THE ERROR NUMBER IS 164
  (1)
  (1)
```

```
 7175
  (1)                                ;*************************************************************************
  (1)                                ;TEST 24:       FSUB (LSI-11 FLOATING SUBTRACT INSTRUCTION)
  (1)                                ;       002460,123456 - 100177,177777 = 002460,123456
  (1)                                ;       PS = 000,       STACK POINTER = SP
  (1)                                ;*************************************************************************
  (1)
  (1)  004672  104400                            SCOPE
  (1)  004674  004567  010312          TST24:    JSR     R5,     PUSHS   ;PUSH 4 WORDS ONTO STACK, SET PRIORITY
  (1)  004700  100177  177777                    .WORD   100177,177777   ;SECOND OPERAND ON TOP
  (1)  004704  002460  123456                    .WORD   002460,123456   ;FIRST OPERAND ON BOTTOM
  (1)  004710  000015                            .WORD   015             ;PROCESSOR PRIORITY LEVEL
  (1)  004712  015634  000340                    .WORD   TRAPER, 340     ;FIS TRAP VECTOR
  (1)
  (1)  004716  000240                            NOP
  (1)  004720  075016                            FSUB    SP              ;FLOATING SUBTRACT ON THE STACK
  (1)
  (1)  004722  004767  010324                    JSR     PC,     POPS    ;POP THE ANSWER
  (1)  004726  022706  000600                    CMP     #BEGIN, SP      ;CHECK THE STACK POINTER
  (1)  004732  001405                            BEQ     TSA24           ;BRANCH IF OK
  (1)  004734  012706  000600                    MOV     #BEGIN, SP      ;RESTORE STACK POINTER
  (1)  004740  104000                            HLT                     ;STACK POINTER FOULED UP
  (3)  004742  000165                            165                     ;THE ERROR NUMBER IS 165
  (1)  004744  000422                            BR      END24           ;SKIP REST OF TEST
  (1)
  (1)  004746  122767  000000  173456  TSA24:    CMPB    #000,   $PSW    ;CHECK PS (EXCEPT T BIT)
  (1)  004754  001402                            BEQ     .+6             ;BRANCH IF OK
  (1)  004756  104000                            HLT                     ;PS NOT EQUAL TO 000
  (3)  004760  000166                            166                     ;THE ERROR NUMBER IS 166
  (1)
  (1)  004762  022767  002460  173446            CMP     #002460,ANS1    ;CHECK FIRST HALF OF ANSWER
  (1)  004770  001402                            BEQ     .+6             ;BRANCH IF OK
  (1)  004772  104002                            HLT+2                   ;ANS1 NOT EQUAL TO 002460
  (3)  004774  000167                            167                     ;THE ERROR NUMBER IS 167
  (1)
  (1)  004776  022767  123456  173434            CMP     #123456,ANS2    ;CHECK SECOND HALF OF ANSWER
  (1)  005004  001402                            BEQ     .+6             ;BRANCH IF OK
  (1)  005006  104002                            HLT+2                   ;ANS2 NOT EQUAL TO 123456
  (3)  005010  000170                            170                     ;THE ERROR NUMBER IS 170
  (1)
  (1)  005012  122767  000024  173364  END24:    CMPB    #24,    $TESTN  ;CHECK THE TEST NUMBER
  (1)  005020  001402                            BEQ     .+6             ;BRANCH IF OK
  (1)  005022  104000                            HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
  (3)  005024  000171                            171                     ;THE ERROR NUMBER IS 171
  (1)
  (1)
```

```
7176
 (1)                                ;*************************************************************************
 (1)                                ;TEST 25:       FSUB (LSI-11 FLOATING SUBTRACT INSTRUCTION)
 (1)                                ;       000425,052525 - 000252,125252 = 000200,000000
 (1)                                ;       PS = 200,       STACK POINTER = R4
 (1)                                ;*************************************************************************
 (1)
 (1)  005026  104400                            SCOPE
 (1)  005030  004567  010330        TST25:      JSR     R5,     PUSHR   ;PUSH 4 WORDS ONTO R4 STACK, SET PRIORITY
 (1)  005034  000252  125252                    .WORD   000252,125252   ;SECOND OPERAND ON TOP
 (1)  005040  000425  052525                    .WORD   000425,052525   ;FIRST OPERAND ON BOTTOM
 (1)  005044  000217                            .WORD   217             ;PROCESSOR PRIORITY LEVEL
 (1)  005046  015634  000340                    .WORD   TRAPER, 340     ;FIS TRAP VECTOR
 (1)  005052  012704  000510                    MOV     #STACK0,R4      ;SET UP STACK POINTER
 (1)
 (1)  005056  000240                            NOP
 (1)  005060  075014                            FSUB    R4              ;FLOATING SUBTRACT ON THE R4 STACK
 (1)
 (1)  005062  004767  010330                    JSR     PC,     POPR    ;POP THE ANSWER
 (1)  005066  010467  173342                    MOV     R4,     $SP     ;SAVE "STACK POINTER"
 (1)  005072  122767  000200  173332           CMPB    #200,   $PSW    ;CHECK PS (EXCEPT T BIT)
 (1)  005100  001402                            BEQ     .+6             ;BRANCH IF OK
 (1)  005102  104000                            HLT                     ;PS NOT EQUAL TO 200
 (3)  005104  000172                            172                     ;THE ERROR NUMBER IS 172
 (1)
 (1)  005106  022767  000514  173320            CMP     #STACK4,$SP     ;CHECK THE STACK POINTER (R4)
 (1)  005114  001402                            BEQ     .+6             ;BRANCH IF OK
 (1)  005116  104000                            HLT                     ;STACK POINTER (R4) NOT EQUAL TO #STACK4
 (3)  005120  000173                            173                     ;THE ERROR NUMBER IS 173
 (1)
 (1)  005122  022767  000200  173306            CMP     #000200,ANS1    ;CHECK FIRST HALF OF ANSWER
 (1)  005130  001402                            BEQ     .+6             ;BRANCH IF OK
 (1)  005132  104002                            HLT+2                   ;ANS1 NOT EQUAL TO 000200
 (3)  005134  000174                            174                     ;THE ERROR NUMBER IS 174
 (1)
 (1)  005136  005767  173276                    TST     ANS2            ;CHECK SECOND HALF OF ANSWER
 (1)  005142  001402                            BEQ     .+6             ;BRANCH IF OK
 (1)  005144  104002                            HLT+2                   ;ANS2 NOT EQUAL TO 000000
 (3)  005146  000175                            175                     ;THE ERROR NUMBER IS 175
 (1)
 (1)  005150  122767  000025  173226  END25:    CMPB    #25,    $TESTN  ;CHECK THE TEST NUMBER
 (1)  005156  001402                            BEQ     .+6             ;BRANCH IF OK
 (1)  005160  104000                            HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
 (3)  005162  000176                            176                     ;THE ERROR NUMBER IS 176
 (1)
 (1)
```

```
 7177
  (1)                                ;*****************************************************************************
  (1)                                ;TEST 26:       FSUB (LSI-11 FLOATING SUBTRACT INSTRUCTION)
  (1)                                ;       177452,125252 - 077652,125252 = 177777,177777
  (1)                                ;       PS = 210,       STACK POINTER = SP
  (1)                                ;*****************************************************************************
  (1)
  (1)  005164  104400                           SCOPE
  (1)  005166  004567  010020        TST26:     JSR     R5,     PUSHS   ;PUSH 4 WORDS ONTO STACK, SET PRIORITY
  (1)  005172  077652  125252                   .WORD   077652,125252   ;SECOND OPERAND ON TOP
  (1)  005176  177452  125252                   .WORD   177452,125252   ;FIRST OPERAND ON BOTTOM
  (1)  005202  000357                           .WORD   357             ;PROCESSOR PRIORITY LEVEL
  (1)  005204  015634  000340                   .WORD   TRAPER, 340     ;FIS TRAP VECTOR
  (1)
  (1)  005210  000240                           NOP
  (1)  005212  075016                           FSUB    SP              ;FLOATING SUBTRACT ON THE STACK
  (1)
  (1)  005214  004767  010032                   JSR     PC,     POPS    ;POP THE ANSWER
  (1)  005220  022706  000600                   CMP     #BEGIN, SP      ;CHECK THE STACK POINTER
  (1)  005224  001405                           BEQ     TSA26           ;BRANCH IF OK
  (1)  005226  012706  000600                   MOV     #BEGIN, SP      ;RESTORE STACK POINTER
  (1)  005232  104000                           HLT                     ;STACK POINTER FOULED UP
  (3)  005234  000177                           177                     ;THE ERROR NUMBER IS 177
  (1)  005236  000422                           BR      END26           ;SKIP REST OF TEST
  (1)
  (1)  005240  122767  000210  173164  TSA26:   CMPB    #210,   $PSW    ;CHECK PS (EXCEPT T BIT)
  (1)  005246  001402                           BEQ     .+6             ;BRANCH IF OK
  (1)  005250  104000                           HLT                     ;PS NOT EQUAL TO 210
  (3)  005252  000200                           200                     ;THE ERROR NUMBER IS 200
  (1)
  (1)  005254  022767  177777  173154           CMP     #177777,ANS1    ;CHECK FIRST HALF OF ANSWER
  (1)  005262  001402                           BEQ     .+6             ;BRANCH IF OK
  (1)  005264  104002                           HLT+2                   ;ANS1 NOT EQUAL TO 177777
  (3)  005266  000201                           201                     ;THE ERROR NUMBER IS 201
  (1)
  (1)  005270  022767  177777  173142           CMP     #177777,ANS2    ;CHECK SECOND HALF OF ANSWER
  (1)  005276  001402                           BEQ     .+6             ;BRANCH IF OK
  (1)  005300  104002                           HLT+2                   ;ANS2 NOT EQUAL TO 177777
  (3)  005302  000202                           202                     ;THE ERROR NUMBER IS 202
  (1)
  (1)  005304  122767  000026  173072  END26:   CMPB    #26,    $TESTN  ;CHECK THE TEST NUMBER
  (1)  005312  001402                           BEQ     .+6             ;BRANCH IF OK
  (1)  005314  104000                           HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
  (3)  005316  000203                           203                     ;THE ERROR NUMBER IS 203
  (1)
  (1)
```

```
 7178
  (1)                                ;*****************************************************************************
  (1)                                ;TEST 27:       FSUB (LSI-11 FLOATING SUBTRACT INSTRUCTION)
  (1)                                ;       035152,125252 - 043125,052525 = 143125,052524
  (1)                                ;       PS = 010,       STACK POINTER = R3
  (1)                                ;*****************************************************************************
  (1)
  (1)  005320  104400                        SCOPE
  (1)  005322  004567  010036        TST27:  JSR     RS,     PUSHR   ;PUSH 4 WORDS ONTO R3 STACK, SET PRIORITY
  (1)  005326  043125  052525                .WORD   043125,052525   ;SECOND OPERAND ON TOP
  (1)  005332  035152  125252                .WORD   035152,125252   ;FIRST OPERAND ON BOTTOM
  (1)  005336  000147                        .WORD   147             ;PROCESSOR PRIORITY LEVEL
  (1)  005340  015634  000340                .WORD   TRAPER, 340     ;FIS TRAP VECTOR
  (1)  005344  012703  000510                MOV     #STACK0,R3      ;SET UP STACK POINTER
  (1)
  (1)  005350  000240                        NOP
  (1)  005352  075013                        FSUB    R3              ;FLOATING SUBTRACT ON THE R3 STACK
  (1)
  (1)  005354  004767  010036                JSR     PC,     POPR    ;POP THE ANSWER
  (1)  005360  010367  173050                MOV     R3,     $SP     ;SAVE "STACK POINTER"
  (1)  005364  122767  000010  173040        CMPB    #010,   $PSW    ;CHECK PS (EXCEPT T BIT)
  (1)  005372  001402                        BEQ     .+6             ;BRANCH IF OK
  (1)  005374  104000                        HLT                     ;PS NOT EQUAL TO 010
  (3)  005376  000204                        204                     ;THE ERROR NUMBER IS 204
  (1)
  (1)  005400  022767  000514  173026        CMP     #STACK4,$SP     ;CHECK THE STACK POINTER (R3)
  (1)  005406  001402                        BEQ     .+6             ;BRANCH IF OK
  (1)  005410  104000                        HLT                     ;STACK POINTER (R3) NOT EQUAL TO #STACK4
  (3)  005412  000205                        205                     ;THE ERROR NUMBER IS 205
  (1)
  (1)  005414  022767  143125  173014        CMP     #143125,ANS1    ;CHECK FIRST HALF OF ANSWER
  (1)  005422  001402                        BEQ     .+6             ;BRANCH IF OK
  (1)  005424  104002                        HLT+2                   ;ANS1 NOT EQUAL TO 143125
  (3)  005426  000206                        206                     ;THE ERROR NUMBER IS 206
  (1)
  (1)  005430  022767  052524  173002        CMP     #052524,ANS2    ;CHECK SECOND HALF OF ANSWER
  (1)  005436  001402                        BEQ     .+6             ;BRANCH IF OK
  (1)  005440  104002                        HLT+2                   ;ANS2 NOT EQUAL TO 052524
  (3)  005442  000207                        207                     ;THE ERROR NUMBER IS 207
  (1)
  (1)  005444  122767  000027  172732  END27: CMPB    #27,    $TESTN  ;CHECK THE TEST NUMBER
  (1)  005452  001402                        BEQ     .+6             ;BRANCH IF OK
  (1)  005454  104000                        HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
  (3)  005456  000210                        210                     ;THE ERROR NUMBER IS 210
  (1)
  (1)
```

```
 7179
  (1)                                ;*****************************************************************************
  (1)                                ;TEST 30:       FSUB (LSI-11 FLOATING SUBTRACT INSTRUCTION)
  (1)                                ;       143325,052525 - 135152,125252 = 143325,052525
  (1)                                ;       PS = 210,       STACK POINTER = R0
  (1)                                ;*****************************************************************************
  (1)
  (1)  005460  104400                          SCOPE
  (1)  005462  004567  007676          TST30:  JSR     R5,     PUSHR   ;PUSH 4 WORDS ONTO R0 STACK, SET PRIORITY
  (1)  005466  135152  125252                  .WORD   135152,125252   ;SECOND OPERAND ON TOP
  (1)  005472  143325  052525                  .WORD   143325,052525   ;FIRST OPERAND ON BOTTOM
  (1)  005476  000243                          .WORD   243             ;PROCESSOR PRIORITY LEVEL
  (1)  005500  015634  000340                  .WORD   TRAPER, 340     ;FIS TRAP VECTOR
  (1)  005504  012700  000510                  MOV     #STACK0,R0      ;SET UP STACK POINTER
  (1)
  (1)  005510  000240                          NOP
  (1)  005512  075010                          FSUB    R0              ;FLOATING SUBTRACT ON THE R0 STACK
  (1)
  (1)  005514  004767  007676                  JSR     PC,     POPR    ;POP THE ANSWER
  (1)  005520  010067  172710                  MOV     R0,     $SP     ;SAVE "STACK POINTER"
  (1)  005524  122767  000210  172700          CMPB    #210,   $PSW    ;CHECK PS (EXCEPT T BIT)
  (1)  005532  001402                          BEQ     .+6             ;BRANCH IF OK
  (1)  005534  104000                          HLT                     ;PS NOT EQUAL TO 210
  (3)  005536  000211                          211                     ;THE ERROR NUMBER IS 211
  (1)
  (1)  005540  022767  000514  172666          CMP     #STACK4,$SP     ;CHECK THE STACK POINTER (R0)
  (1)  005546  001402                          BEQ     .+6             ;BRANCH IF OK
  (1)  005550  104000                          HLT                     ;STACK POINTER (R0) NOT EQUAL TO #STACK4
  (3)  005552  000212                          212                     ;THE ERROR NUMBER IS 212
  (1)
  (1)  005554  022767  143325  172654          CMP     #143325,ANS1    ;CHECK FIRST HALF OF ANSWER
  (1)  005562  001402                          BEQ     .+6             ;BRANCH IF OK
  (1)  005564  104002                          HLT+2                   ;ANS1 NOT EQUAL TO 143325
  (3)  005566  000213                          213                     ;THE ERROR NUMBER IS 213
  (1)
  (1)  005570  022767  052525  172642          CMP     #052525,ANS2    ;CHECK SECOND HALF OF ANSWER
  (1)  005576  001402                          BEQ     .+6             ;BRANCH IF OK
  (1)  005600  104002                          HLT+2                   ;ANS2 NOT EQUAL TO 052525
  (3)  005602  000214                          214                     ;THE ERROR NUMBER IS 214
  (1)
  (1)  005604  122767  000030  172572  END30: CMPB    #30,    $TESTN  ;CHECK THE TEST NUMBER
  (1)  005612  001402                          BEQ     .+6             ;BRANCH IF OK
  (1)  005614  104000                          HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
  (3)  005616  000215                          215                     ;THE ERROR NUMBER IS 215
  (1)
  (1)
```

```
7180
 (1)                                ;*************************************************************************
 (1)                                ;TEST 31:       FSUB (LSI-11 FLOATING SUBTRACT INSTRUCTION)
 (1)                                ;       135152,125252 - 143325,052525 = 043325,052525
 (1)                                ;       PS = 200,       STACK POINTER = R5
 (1)                                ;*************************************************************************
 (1)
 (1)  005620  104400                            SCOPE
 (1)  005622  004567  007536        TST31:      JSR     R5,     PUSHR   ;PUSH 4 WORDS ONTO R5 STACK, SET PRIORITY
 (1)  005626  143325  052525                    .WORD   143325,052525   ;SECOND OPERAND ON TOP
 (1)  005632  135152  125252                    .WORD   135152,125252   ;FIRST OPERAND ON BOTTOM
 (1)  005636  000357                            .WORD   357             ;PROCESSOR PRIORITY LEVEL
 (1)  005640  015634  000340                    .WORD   TRAPER, 340     ;FIS TRAP VECTOR
 (1)  005644  012705  000510                    MOV     #STACK0,R5      ;SET UP STACK POINTER
 (1)
 (1)  005650  000240                            NOP
 (1)  005652  075015                            FSUB    R5              ;FLOATING SUBTRACT ON THE R5 STACK
 (1)
 (1)  005654  004767  007536                    JSR     PC,     POPR    ;POP THE ANSWER
 (1)  005660  010567  172550                    MOV     R5,     $SP     ;SAVE "STACK POINTER"
 (1)  005664  122767  000200  172540           CMPB    #200,   $PSW    ;CHECK PS (EXCEPT T BIT)
 (1)  005672  001402                            BEQ     .+6             ;BRANCH IF OK
 (1)  005674  104000                            HLT                     ;PS NOT EQUAL TO 200
 (3)  005676  000216                            216                     ;THE ERROR NUMBER IS 216
 (1)
 (1)  005700  022767  000514  172526            CMP     #STACK4,$SP     ;CHECK THE STACK POINTER (R5)
 (1)  005706  001402                            BEQ     .+6             ;BRANCH IF OK
 (1)  005710  104000                            HLT                     ;STACK POINTER (R5) NOT EQUAL TO #STACK4
 (3)  005712  000217                            217                     ;THE ERROR NUMBER IS 217
 (1)
 (1)  005714  022767  043325  172514            CMP     #043325,ANS1    ;CHECK FIRST HALF OF ANSWER
 (1)  005722  001402                            BEQ     .+6             ;BRANCH IF OK
 (1)  005724  104002                            HLT+2                   ;ANS1 NOT EQUAL TO 043325
 (3)  005726  000220                            220                     ;THE ERROR NUMBER IS 220
 (1)
 (1)  005730  022767  052525  172502            CMP     #052525,ANS2    ;CHECK SECOND HALF OF ANSWER
 (1)  005736  001402                            BEQ     .+6             ;BRANCH IF OK
 (1)  005740  104002                            HLT+2                   ;ANS2 NOT EQUAL TO 052525
 (3)  005742  000221                            221                     ;THE ERROR NUMBER IS 221
 (1)
 (1)  005744  122767  000031  172432  END31:    CMPB    #31,    $TESTN  ;CHECK THE TEST NUMBER
 (1)  005752  001402                            BEQ     .+6             ;BRANCH IF OK
 (1)  005754  104000                            HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
 (3)  005756  000222                            222                     ;THE ERROR NUMBER IS 222
 (1)
 (1)
```

```
 7181
  (1)                                ;*************************************************************************
  (1)                                ;TEST 32:       FSUB (LSI-11 FLOATING SUBTRACT INSTRUCTION)
  (1)                                ;       043125,052525 - 035152,125252 = 043125,052524
  (1)                                ;       PS = 000,       STACK POINTER = R2
  (1)                                ;*************************************************************************
  (1)
  (1)  005760  104400                          SCOPE
  (1)  005762  004567  007376        TST32:    JSR     R5,     PUSHR   ;PUSH 4 WORDS ONTO R2 STACK, SET PRIORITY
  (1)  005766  035152  125252                  .WORD   035152,125252   ;SECOND OPERAND ON TOP
  (1)  005772  043125  052525                  .WORD   043125,052525   ;FIRST OPERAND ON BOTTOM
  (1)  005776  000040                          .WORD   040             ;PROCESSOR PRIORITY LEVEL
  (1)  006000  015634  000340                  .WORD   TRAPER, 340     ;FIS TRAP VECTOR
  (1)  006004  012702  000510                  MOV     #STACK0,R2      ;SET UP STACK POINTER
  (1)
  (1)  006010  000240                          NOP
  (1)  006012  075012                          FSUB    R2              ;FLOATING SUBTRACT ON THE R2 STACK
  (1)
  (1)  006014  004767  007376                  JSR     PC,     POPR    ;POP THE ANSWER
  (1)  006020  010267  172410                  MOV     R2,     $SP     ;SAVE "STACK POINTER"
  (1)  006024  105767  172402                  TSTB    $PSW            ;CHECK PS (EXCEPT T BIT)
  (1)  006030  001402                          BEQ     .+6             ;BRANCH IF OK
  (1)  006032  104000                          HLT                     ;PS NOT EQUAL TO 000
  (3)  006034  000223                          223                     ;THE ERROR NUMBER IS 223
  (1)
  (1)  006036  022767  000514  172370          CMP     #STACK4,$SP     ;CHECK THE STACK POINTER (R2)
  (1)  006044  001402                          BEQ     .+6             ;BRANCH IF OK
  (1)  006046  104000                          HLT                     ;STACK POINTER (R2) NOT EQUAL TO #STACK4
  (3)  006050  000224                          224                     ;THE ERROR NUMBER IS 224
  (1)
  (1)  006052  022767  043125  172356          CMP     #043125,ANS1    ;CHECK FIRST HALF OF ANSWER
  (1)  006060  001402                          BEQ     .+6             ;BRANCH IF OK
  (1)  006062  104002                          HLT+2                   ;ANS1 NOT EQUAL TO 043125
  (3)  006064  000225                          225                     ;THE ERROR NUMBER IS 225
  (1)
  (1)  006066  022767  052524  172344          CMP     #052524,ANS2    ;CHECK SECOND HALF OF ANSWER
  (1)  006074  001402                          BEQ     .+6             ;BRANCH IF OK
  (1)  006076  104002                          HLT+2                   ;ANS2 NOT EQUAL TO 052524
  (3)  006100  000226                          226                     ;THE ERROR NUMBER IS 226
  (1)
  (1)  006102  122767  000032  172274  END32:  CMPB    #32,    $TESTN  ;CHECK THE TEST NUMBER
  (1)  006110  001402                          BEQ     .+6             ;BRANCH IF OK
  (1)  006112  104000                          HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
  (3)  006114  000227                          227                     ;THE ERROR NUMBER IS 227
  (1)
  (1)
```

```
 7185
  (1)                                ;*****************************************************************************
  (1)                                ;TEST 33:       FSUB (LSI-11 FLOATING SUBTRACT INSTRUCTION)
  (1)                                ;       000425,052525 - 000252,125253 ==> UNDERFLOW
  (1)                                ;       PS(ON STACK) = 212,     STACK POINTER = SP
  (1)                                ;*****************************************************************************
  (1)
  (1)  006116  104400                           SCOPE
  (1)  006120  004567  007066        TST33:     JSR     R5,     PUSHS   ;PUSH 4 WORDS ONTO STACK, SET PRIORITY
  (1)  006124  000252  125253                   .WORD   000252,125253   ;SECOND OPERAND ON TOP
  (1)  006130  000425  052525                   .WORD   000425,052525   ;FIRST OPERAND ON BOTTOM
  (1)  006134  000257                           .WORD   257             ;PROCESSOR PRIORITY LEVEL
  (1)  006136  006164  000340                   .WORD   ISR33,  340     ;FIS TRAP VECTOR
  (1)
  (1)  006142  000240                           NOP
  (1)  006144  075016                           FSUB    SP              ;FLOATING SUBTRACT ON THE STACK
  (1)
  (1)  006146  004767  007100        RTA33:     JSR     PC,     POPS    ;POP THE "ANSWER"
  (1)  006152  104002                           HLT+2                   ;FIS TRAP DIDN'T OCCURE!
  (3)  006154  000230                           230                     ;THE ERROR NUMBER IS 230
  (1)  006156  012706  000600                   MOV     #BEGIN, SP      ;RESTORE THE STACK POINTER
  (1)  006162  000464                           BR      END33
  (1)
  (1)  006164  004767  007114        ISR33:     JSR     PC,     POPES   ;POP ALL DATA OFF THE STACK
  (1)  006170  022706  000600                   CMP     #BEGIN, SP      ;CHECK THE STACK POINTER
  (1)  006174  001405                           BEQ     ISA33           ;BRANCH IF OK
  (1)  006176  012706  000600                   MOV     #BEGIN, SP      ;RESTORE THE STACK POINTER
  (1)  006202  104000                           HLT                     ;STACK POINTER FOULED UP
  (3)  006204  000231                           231                     ;THE ERROR NUMBER IS 231
  (1)  006206  000452                           BR      END33           ;SKIP REST OF TEST
  (1)
  (1)  006210  122767  000340  172214  ISA33:   CMPB    #340,   $PSW    ;CHECK PS AFTER FIS TRAP
  (1)  006216  001402                           BEQ     .+6             ;BRANCH IF OK
  (1)  006220  104000                           HLT                     ;PS AFTER FIS TRAP NOT EQUAL TO 340
  (3)  006222  000232                           232                     ;THE ERROR NUMBER IS 232
  (1)
  (1)  006224  022767  006146  172204           CMP     #RTA33, ANS1    ;CHECK FIS TRAP RETURN ADDRESS
  (1)  006232  001402                           BEQ     .+6             ;BRANCH IF OK
  (1)  006234  104001                           HLT+1                   ;FIS TRAP AT WRONG ADDRESS
  (3)  006236  000233                           233                     ;THE ERROR NUMBER IS 233
  (1)
  (1)  006240  022767  000212  172172           CMP     #212,   ANS2    ;CHECK PS BEFORE FIS TRAP
  (1)  006246  001402                           BEQ     .+6             ;BRANCH IF OK
  (1)  006250  104002                           HLT+2                   ;PS AT FIS TRAP TIME NOT 212
  (3)  006252  000234                           234                     ;THE ERROR NUMBER IS 234
  (1)
  (1)  006254  022767  000252  172160           CMP     #000252,ANS3    ;CHECK DATA FROM THE STACK
  (1)  006262  001402                           BEQ     .+6             ;BRANCH IF OK
  (1)  006264  104004                           HLT+4                   ;DATA ON STACK (000252) CHANGED
  (3)  006266  000235                           235                     ;THE ERROR NUMBER IS 235
  (1)
  (1)  006270  022767  125253  172146           CMP     #125253,ANS4    ;CHECK DATA FROM STACK
  (1)  006276  001402                           BEQ     .+6             ;BRANCH IF OK
  (1)  006300  104004                           HLT+4                   ;DATA ON STACK (125253) CHANGED
  (3)  006302  000236                           236                     ;THE ERROR NUMBER IS 236
  (1)
```

```
    (1)  006304  022767  000425  172134          CMP     #000425,ANS5    ;CHECK DATA FROM STACK
    (1)  006312  001402                          BEQ     .+6             ;BRANCH IF OK
    (1)  006314  104006                          HLT+6                   ;DATA ON STACK (000425) CHANGED
    (3)  006316  000237                          237                     ;THE ERROR NUMBER IS 237
    (1)
    (1)  006320  022767  052525  172122          CMP     #052525,ANS6    ;CHECK DATA FROM STACK
    (1)  006326  001402                          BEQ     .+6             ;BRANCH IF OK
    (1)  006330  104006                          HLT+6                   ;DATA ON STACK (052525) CHANGED
    (3)  006332  000240                          240                     ;THE ERROR NUMBER IS 240
    (1)
    (1)  006334  122767  000033  172042  END33:  CMPB    #33,    $TESTN  ;CHECK THE TEST NUMBER
    (1)  006342  001402                          BEQ     .+6             ;BRANCH IF OK
    (1)  006344  104000                          HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
    (3)  006346  000241                          241                     ;THE ERROR NUMBER IS 241
    (1)
    (1)
```

```
7189
 (1)                                ;*************************************************************************
 (1)                                ;TEST 34:       FSUB (LSI-11 FLOATING SUBTRACT INSTRUCTION)
 (1)                                ;       077652,125253 - 177452,125252 ==> OVERFLOW
 (1)                                ;       PS(ON STACK) = 002,     STACK POINTER = R3
 (1)                                ;*************************************************************************
 (1)
 (1)  006350  104400                        SCOPE
 (1)  006352  004567  007006        TST34:  JSR     R5,     PUSHR   ;PUSH 4 WORDS ONTO R3 STACK, SET PRIORITY
 (1)  006356  177452  125252                .WORD   177452,125252   ;SECOND OPERAND ON TOP
 (1)  006362  077652  125253                .WORD   077652,125253   ;FIRST OPERAND ON BOTTOM
 (1)  006366  000015                        .WORD   015             ;PROCESSOR PRIORITY LEVEL
 (1)  006370  006422  000344                .WORD   ISR34,  344     ;FIS TRAP VECTOR
 (1)  006374  012703  000510                MOV     #STACKO,R3      ;SET UP R3 AS STACK POINTER
 (1)
 (1)  006400  000240                        NOP
 (1)  006402  075013                        FSUB    R3              ;FLOATING SUBTRACT ON THE R3 STACK
 (1)
 (1)  006404  004767  007006        RTA34:  JSR     PC,     POPR    ;POP THE "ANSWER"
 (1)  006410  010367  172020                MOV     R3,     $SP     ;SAVE STACK POINTER (R3)
 (1)  006414  104002                        HLT+2                   ;FIS TRAP DIDN'T OCCURE!
 (3)  006416  000242                        242                     ;THE ERROR NUMBER IS 242
 (1)  006420  000464                        BR      END34
 (1)
 (1)  006422  004767  007020        ISR34:  JSR     PC,     POPER   ;POP ALL DATA OFF THE STACKS
 (1)  006426  010367  172002                MOV     R3,     $SP     ;SAVE STACK POINTER (R3)
 (1)  006432  122767  000344  171772        CMPB    #344,   $PSW    ;CHECK PS AFTER FIS TRAP
 (1)  006440  001402                        BEQ     .+6             ;BRANCH IF OK
 (1)  006442  104000                        HLT                     ;PS AFTER FIS TRAP NOT EQUAL TO 344
 (3)  006444  000243                        243                     ;THE ERROR NUMBER IS 243
 (1)
 (1)  006446  022767  000510  171760        CMP     #STACKO,$SP     ;CHECK THE STACK POINTER (R3)
 (1)  006454  001402                        BEQ     .+6             ;BRANCH IF OK
 (1)  006456  104000                        HLT                     ;STACK POINTER (R3) NOT EQUAL TO #STACKO
 (3)  006460  000244                        244                     ;THE ERROR NUMBER IS 244
 (1)
 (1)  006462  022767  006404  171746        CMP     #RTA34, ANS1    ;CHECK FIS TRAP RETURN ADDRESS
 (1)  006470  001402                        BEQ     .+6             ;BRANCH IF OK
 (1)  006472  104001                        HLT+1                   ;FIS TRAP AT WRONG ADDRESS
 (3)  006474  000245                        245                     ;THE ERROR NUMBER IS 245
 (1)
 (1)  006476  022767  000002  171734        CMP     #002,   ANS2    ;CHECK PS BEFORE FIS TRAP
 (1)  006504  001402                        BEQ     .+6             ;BRANCH IF OK
 (1)  006506  104002                        HLT+2                   ;PS AT FIS TRAP TIME NOT 002
 (3)  006510  000246                        246                     ;THE ERROR NUMBER IS 246
 (1)
 (1)  006512  022767  177452  171722        CMP     #177452,ANS3    ;CHECK DATA FROM THE STACK
 (1)  006520  001402                        BEQ     .+6             ;BRANCH IF OK
 (1)  006522  104004                        HLT+4                   ;DATA ON STACK (177452) CHANGED
 (3)  006524  000247                        247                     ;THE ERROR NUMBER IS 247
 (1)
 (1)  006526  022767  125252  171710        CMP     #125252,ANS4    ;CHECK DATA FROM STACK
 (1)  006534  001402                        BEQ     .+6             ;BRANCH IF OK
 (1)  006536  104004                        HLT+4                   ;DATA ON STACK (125252) CHANGED
 (3)  006540  000250                        250                     ;THE ERROR NUMBER IS 250
 (1)
```

```
(1)  006542  022767  077652  171676          CMP     #077652,ANS5    ;CHECK DATA FROM STACK
(1)  006550  001402                          BEQ     .+6             ;BRANCH IF OK
(1)  006552  104006                          HLT+6                   ;DATA ON STACK (077652) CHANGED
(3)  006554  000251                          251                     ;THE ERROR NUMBER IS 251
(1)
(1)  006556  022767  125253  171664          CMP     #125253,ANS6    ;CHECK DATA FROM STACK
(1)  006564  001402                          BEQ     .+6             ;BRANCH IF OK
(1)  006566  104006                          HLT+6                   ;DATA ON STACK (125253) CHANGED
(3)  006570  000252                          252                     ;THE ERROR NUMBER IS 252
(1)
(1)  006572  122767  000034  171604  END34:  CMPB    #34,    STESTN  ;CHECK THE TEST NUMBER
(1)  006600  001402                          BEQ     .+6             ;BRANCH IF OK
(1)  006602  104000                          HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
(3)  006604  000253                          253                     ;THE ERROR NUMBER IS 253
(1)
(1)
```

```
 7193
  (1)                                ;*************************************************************************
  (1)                                ;TEST 35:       FMUL (LSI-11 FLOATING MULTIPLY INSTRUCTION)
  (1)                                ;       000000,000000 * 000000,000000 = 000000,000000
  (1)                                ;       PS = 004,       STACK POINTER = R4
  (1)                                ;*************************************************************************
  (1)
  (1)  006606  104400                          SCOPE
  (1)  006610  004567  006550          TST35:  JSR     R5,     PUSHR   ;PUSH 4 WORDS ONTO R4 STACK, SET PRIORITY
  (1)  006614  000000  000000                  .WORD   000000,000000   ;SECOND OPERAND ON TOP
  (1)  006620  000000  000000                  .WORD   000000,000000   ;FIRST OPERAND ON BOTOM
  (1)  006624  000111                          .WORD   111             ;PROCESSOR PRIORITY LEVEL
  (1)  006626  015634  000340                  .WORD   TRAPER, 340     ;FIS TRAP VECTOR
  (1)  006632  012704  000510                  MOV     #STACK0,R4      ;SET UP STACK POINTER
  (1)
  (1)  006636  000240                          NOP
  (1)  006640  075024                          FMUL    R4              ;FLOATING MULTIPLY ON THE R4 STACK
  (1)
  (1)  006642  004767  006550                  JSR     PC,     POPR    ;POP THE ANSWER
  (1)  006646  010467  171562                  MOV     R4,     $SP     ;SAVE "STACK POINTER"
  (1)  006652  122767  000004  171552          CMPB    #004,   $PSW    ;CHECK PS (EXCEPT T BIT)
  (1)  006660  001402                          BEQ     .+6             ;BRANCH IF OK
  (1)  006662  104000                          HLT                     ;PS NOT EQUAL TO 004
  (3)  006664  000254                          254                     ;THE ERROR NUMBER IS 254
  (1)
  (1)  006666  022767  000514  171540          CMP     #STACK4,$SP     ;CHECK THE STACK POINTER (R4)
  (1)  006674  001402                          BEQ     .+6             ;BRANCH IF OK
  (1)  006676  104000                          HLT                     ;STACK POINTER (R4) NOT EQUAL TO #STACK4
  (3)  006700  000255                          255                     ;THE ERROR NUMBER IS 255
  (1)
  (1)  006702  005767  171530                  TST     ANS1            ;CHECK FIRST HALF OF ANSWER
  (1)  006706  001402                          BEQ     .+6             ;BRANCH IF OK
  (1)  006710  104002                          HLT+2                   ;ANS1 NOT EQUAL TO 000000
  (3)  006712  000256                          256                     ;THE ERROR NUMBER IS 256
  (1)
  (1)  006714  005767  171520                  TST     ANS2            ;CHECK SECOND HALF OF ANSWER
  (1)  006720  001402                          BEQ     .+6             ;BRANCH IF OK
  (1)  006722  104002                          HLT+2                   ;ANS2 NOT EQUAL TO 000000
  (3)  006724  000257                          257                     ;THE ERROR NUMBER IS 257
  (1)
  (1)  006726  122767  000035  171450  END35:  CMPB    #35,    $TESTN  ;CHECK THE TEST NUMBER
  (1)  006734  001402                          BEQ     .+6             ;BRANCH IF OK
  (1)  006736  104000                          HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
  (3)  006740  000260                          260                     ;THE ERROR NUMBER IS 260
  (1)
  (1)
```

```
 7194
  (1)                                ;*****************************************************************************
  (1)                                ;TEST 36:       FMUL (LSI-11 FLOATING MULTIPLY INSTRUCTION)
  (1)                                ;       140200,000000 * 052345,123456 = 152345,123456
  (1)                                ;       PS = 210,       STACK POINTER = R2
  (1)                                ;*****************************************************************************
  (1)
  (1)  006742  104400                           SCOPE
  (1)  006744  004567  006414        TST36:     JSR     R5,     PUSHR   ;PUSH 4 WORDS ONTO R2 STACK, SET PRIORITY
  (1)  006750  052345  123456                   .WORD   052345,123456   ;SECOND OPERAND ON TOP
  (1)  006754  140200  000000                   .WORD   140200,000000   ;FIRST OPERAND ON BOTOM
  (1)  006760  000343                           .WORD   343             ;PROCESSOR PRIORITY LEVEL
  (1)  006762  015634  000340                   .WORD   TRAPER, 340     ;FIS TRAP VECTOR
  (1)  006766  012702  000510                   MOV     #STACK0,R2      ;SET UP STACK POINTER
  (1)
  (1)  006772  000240                           NOP
  (1)  006774  075022                           FMUL    R2              ;FLOATING MULTIPLY ON THE R2 STACK
  (1)
  (1)  006776  004767  006414                   JSR     PC,     POPR    ;POP THE ANSWER
  (1)  007002  010267  171426                   MOV     R2,     $SP     ;SAVE "STACK POINTER"
  (1)  007006  122767  000210  171416           CMPB    #210,   $PSW    ;CHECK PS (EXCEPT T BIT)
  (1)  007014  001402                           BEQ     .+6             ;BRANCH IF OK
  (1)  007016  104000                           HLT                     ;PS NOT EQUAL TO 210
  (3)  007020  000261                           261                     ;THE ERROR NUMBER IS 261
  (1)
  (1)  007022  022767  000514  171404           CMP     #STACK4,$SP     ;CHECK THE STACK POINTER (R2)
  (1)  007030  001402                           BEQ     .+6             ;BRANCH IF OK
  (1)  007032  104000                           HLT                     ;STACK POINTER (R2) NOT EQUAL TO #STACK4
  (3)  007034  000262                           262                     ;THE ERROR NUMBER IS 262
  (1)
  (1)  007036  022767  152345  171372           CMP     #152345,ANS1    ;CHECK FIRST HALF OF ANSWER
  (1)  007044  001402                           BEQ     .+6             ;BRANCH IF OK
  (1)  007046  104002                           HLT+2                   ;ANS1 NOT EQUAL TO 152345
  (3)  007050  000263                           263                     ;THE ERROR NUMBER IS 263
  (1)
  (1)  007052  022767  123456  171360           CMP     #123456,ANS2    ;CHECK SECOND HALF OF ANSWER
  (1)  007060  001402                           BEQ     .+6             ;BRANCH IF OK
  (1)  007062  104002                           HLT+2                   ;ANS2 NOT EQUAL TO 123456
  (3)  007064  000264                           264                     ;THE ERROR NUMBER IS 264
  (1)
  (1)  007066  122767  000036  171310  END36:   CMPB    #36,    $TESTN  ;CHECK THE TEST NUMBER
  (1)  007074  001402                           BEQ     .+6             ;BRANCH IF OK
  (1)  007076  104000                           HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
  (3)  007100  000265                           265                     ;THE ERROR NUMBER IS 265
  (1)
  (1)
```

```
 7195
  (1)                                ;*************************************************************************
  (1)                                ;TEST 37:       FMUL (LSI-11 FLOATING MULTIPLY INSTRUCTION)
  (1)                                ;       100125,052525 * 135753,024642 = 000000,000000
  (1)                                ;       PS = 004,       STACK POINTER = R5
  (1)                                ;*************************************************************************
  (1)
  (1)  007102  104400                           SCOPE
  (1)  007104  004567  006254        TST37:     JSR     R5,     PUSHR   ;PUSH 4 WORDS ONTO R5 STACK, SET PRIORITY
  (1)  007110  135753  024642                   .WORD   135753,024642   ;SECOND OPERAND ON TOP
  (1)  007114  100125  052525                   .WORD   100125,052525   ;FIRST OPERAND ON BOTOM
  (1)  007120  000117                           .WORD   117             ;PROCESSOR PRIORITY LEVEL
  (1)  007122  015634  000340                   .WORD   TRAPER, 340     ;FIS TRAP VECTOR
  (1)  007126  012705  000510                   MOV     #STACK0,R5      ;SET UP STACK POINTER
  (1)
  (1)  007132  000240                           NOP
  (1)  007134  075025                           FMUL    R5              ;FLOATING MULTIPLY ON THE R5 STACK
  (1)
  (1)  007136  004767  006254                   JSR     PC,     POPR    ;POP THE ANSWER
  (1)  007142  010567  171266                   MOV     R5,     $SP     ;SAVE "STACK POINTER"
  (1)  007146  122767  000004  171256           CMPB    #004,   $PSW    ;CHECK PS (EXCEPT T BIT)
  (1)  007154  001402                           BEQ     .+6             ;BRANCH IF OK
  (1)  007156  104000                           HLT                     ;PS NOT EQUAL TO 004
  (3)  007160  000266                           266                     ;THE ERROR NUMBER IS 266
  (1)
  (1)  007162  022767  000514  171244           CMP     #STACK4,$SP     ;CHECK THE STACK POINTER (R5)
  (1)  007170  001402                           BEQ     .+6             ;BRANCH IF OK
  (1)  007172  104000                           HLT                     ;STACK POINTER (R5) NOT EQUAL TO #STACK4
  (3)  007174  000267                           267                     ;THE ERROR NUMBER IS 267
  (1)
  (1)  007176  005767  171234                   TST     ANS1            ;CHECK FIRST HALF OF ANSWER
  (1)  007202  001402                           BEQ     .+6             ;BRANCH IF OK
  (1)  007204  104002                           HLT+2                   ;ANS1 NOT EQUAL TO 000000
  (3)  007206  000270                           270                     ;THE ERROR NUMBER IS 270
  (1)
  (1)  007210  005767  171224                   TST     ANS2            ;CHECK SECOND HALF OF ANSWER
  (1)  007214  001402                           BEQ     .+6             ;BRANCH IF OK
  (1)  007216  104002                           HLT+2                   ;ANS2 NOT EQUAL TO 000000
  (3)  007220  000271                           271                     ;THE ERROR NUMBER IS 271
  (1)
  (1)  007222  122767  000037  171154  END37:   CMPB    #37,    $TESTN  ;CHECK THE TEST NUMBER
  (1)  007230  001402                           BEQ     .+6             ;BRANCH IF OK
  (1)  007232  104000                           HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
  (3)  007234  000272                           272                     ;THE ERROR NUMBER IS 272
  (1)
  (1)
```

```
7196
 (1)                                ;*****************************************************************************
 (1)                                ;TEST 40:       FMUL (LSI-11 FLOATING MULTIPLY INSTRUCTION)
 (1)                                ;       161616,161616 * 000052,125252 = 000000,000000
 (1)                                ;       PS = 204,       STACK POINTER = R3
 (1)                                ;*****************************************************************************
 (1)
 (1)  007236  104400                            SCOPE
 (1)  007240  004567  006120        TST40:      JSR     R5,     PUSHR   ;PUSH 4 WORDS ONTO R3 STACK, SET PRIORITY
 (1)  007244  000052  125252                    .WORD   000052,125252   ;SECOND OPERAND ON TOP
 (1)  007250  161616  161616                    .WORD   161616,161616   ;FIRST OPERAND ON BOTOM
 (1)  007254  000217                            .WORD   217             ;PROCESSOR PRIORITY LEVEL
 (1)  007256  015634  000340                    .WORD   TRAPER, 340     ;FIS TRAP VECTOR
 (1)  007262  012703  000510                    MOV     #STACK0,R3      ;SET UP STACK POINTER
 (1)
 (1)  007266  000240                            NOP
 (1)  007270  075023                            FMUL    R3              ;FLOATING MULTIPLY ON THE R3 STACK
 (1)
 (1)  007272  004767  006120                    JSR     PC,     POPR    ;POP THE ANSWER
 (1)  007276  010367  171132                    MOV     R3,     $SP     ;SAVE "STACK POINTER"
 (1)  007302  122767  000204  171122            CMPB    #204,   SPSW    ;CHECK PS (EXCEPT T BIT)
 (1)  007310  001402                            BEQ     .+6             ;BRANCH IF OK
 (1)  007312  104000                            HLT                     ;PS NOT EQUAL TO 204
 (3)  007314  000273                            273                     ;THE ERROR NUMBER IS 273
 (1)
 (1)  007316  022767  000514  171110            CMP     #STACK4,$SP     ;CHECK THE STACK POINTER (R3)
 (1)  007324  001402                            BEQ     .+6             ;BRANCH IF OK
 (1)  007326  104000                            HLT                     ;STACK POINTER (R3) NOT EQUAL TO #STACK4
 (3)  007330  000274                            274                     ;THE ERROR NUMBER IS 274
 (1)
 (1)  007332  005767  171100                    TST     ANS1            ;CHECK FIRST HALF OF ANSWER
 (1)  007336  001402                            BEQ     .+6             ;BRANCH IF OK
 (1)  007340  104002                            HLT+2                   ;ANS1 NOT EQUAL TO 000000
 (3)  007342  000275                            275                     ;THE ERROR NUMBER IS 275
 (1)
 (1)  007344  005767  171070                    TST     ANS2            ;CHECK SECOND HALF OF ANSWER
 (1)  007350  001402                            BEQ     .+6             ;BRANCH IF OK
 (1)  007352  104002                            HLT+2                   ;ANS2 NOT EQUAL TO 000000
 (3)  007354  000276                            276                     ;THE ERROR NUMBER IS 276
 (1)
 (1)  007356  122767  000040  171020  END40:    CMPB    #40,    $TESTN  ;CHECK THE TEST NUMBER
 (1)  007364  001402                            BEQ     .+6             ;BRANCH IF OK
 (1)  007366  104000                            HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
 (3)  007370  000277                            277                     ;THE ERROR NUMBER IS 277
 (1)
 (1)
```

```
7197
 (1)                                ;*****************************************************************************
 (1)                                ;TEST 41:       FMUL (LSI-11 FLOATING MULTIPLY INSTRUCTION)
 (1)                                ;       176452,125252 * 041500,000000 = 177777,177777
 (1)                                ;       PS = 210,       STACK POINTER = SP
 (1)                                ;*****************************************************************************
 (1)
 (1)  007372  104400                            SCOPE
 (1)  007374  004567  005612        TST41:      JSR     R5,     PUSHS   ;PUSH 4 WORDS ONTO STACK, SET PRIORITY
 (1)  007400  041500  000000                    .WORD   041500,000000   ;SECOND OPERAND ON TOP
 (1)  007404  176452  125252                    .WORD   176452,125252   ;FIRST OPERAND ON BOTTOM
 (1)  007410  000357                            .WORD   357             ;PROCESSOR PRIORITY LEVEL
 (1)  007412  015634  000340                    .WORD   TRAPER, 340     ;FIS TRAP VECTOR
 (1)
 (1)  007416  000240                            NOP
 (1)  007420  075026                            FMUL    SP              ;FLOATING MULTIPLY ON THE STACK
 (1)
 (1)  007422  004767  005624                    JSR     PC,     POPS    ;POP THE ANSWER
 (1)  007426  022706  000600                    CMP     #BEGIN, SP      ;CHECK THE STACK POINTER
 (1)  007432  001405                            BEQ     TSA41           ;BRANCH IF OK
 (1)  007434  012706  000600                    MOV     #BEGIN, SP      ;RESTORE STACK POINTER
 (1)  007440  104000                            HLT                     ;STACK POINTER FOULED UP
 (3)  007442  000300                            300                     ;THE ERROR NUMBER IS 300
 (1)  007444  000422                            BR      END41           ;SKIP REST OF TEST
 (1)
 (1)  007446  122767  000210  170756  TSA41:    CMPB    #210,   $PSW    ;CHECK PS (EXCEPT T BIT)
 (1)  007454  001402                            BEQ     .+6             ;BRANCH IF OK
 (1)  007456  104000                            HLT                     ;PS NOT EQUAL TO 210
 (3)  007460  000301                            301                     ;THE ERROR NUMBER IS 301
 (1)
 (1)  007462  022767  177777  170746            CMP     #177777,ANS1    ;CHECK FIRST HALF OF ANSWER
 (1)  007470  001402                            BEQ     .+6             ;BRANCH IF OK
 (1)  007472  104002                            HLT+2                   ;ANS1 NOT EQUAL TO 177777
 (3)  007474  000302                            302                     ;THE ERROR NUMBER IS 302
 (1)
 (1)  007476  022767  177777  170734            CMP     #177777,ANS2    ;CHECK SECOND HALF OF ANSWER
 (1)  007504  001402                            BEQ     .+6             ;BRANCH IF OK
 (1)  007506  104002                            HLT+2                   ;ANS2 NOT EQUAL TO 177777
 (3)  007510  000303                            303                     ;THE ERROR NUMBER IS 303
 (1)
 (1)  007512  122767  000041  170664  END41:    CMPB    #41,    $TESTN  ;CHECK THE TEST NUMBER
 (1)  007520  001402                            BEQ     .+6             ;BRANCH IF OK
 (1)  007522  104000                            HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
 (3)  007524  000304                            304                     ;THE ERROR NUMBER IS 304
 (1)
 (1)
```

```
 7198
  (1)                                ;*****************************************************************************
  (1)                                ;TEST 42:       FMUL (LSI-11 FLOATING MULTIPLY INSTRUCTION)
  (1)                                ;       124252,125252 * 114100,000001 = 000200,000000
  (1)                                ;       PS = 200,       STACK POINTER = R1
  (1)                                ;*****************************************************************************
  (1)
  (1)  007526  104400                           SCOPE
  (1)  007530  004567  005630        TST42:     JSR     R5,     PUSHR   ;PUSH 4 WORDS ONTO R1 STACK, SET PRIORITY
  (1)  007534  114100  000001                   .WORD   114100,000001   ;SECOND OPERAND ON TOP
  (1)  007540  124252  125252                   .WORD   124252,125252   ;FIRST OPERAND ON BOTOM
  (1)  007544  000200                           .WORD   200             ;PROCESSOR PRIORITY LEVEL
  (1)  007546  015634  000340                   .WORD   TRAPER, 340     ;FIS TRAP VECTOR
  (1)  007552  012701  000510                   MOV     #STACK0,R1      ;SET UP STACK POINTER
  (1)
  (1)  007556  000240                           NOP
  (1)  007560  075021                           FMUL    R1              ;FLOATING MULTIPLY ON THE R1 STACK
  (1)
  (1)  007562  004767  005630                   JSR     PC,     POPR    ;POP THE ANSWER
  (1)  007566  010167  170642                   MOV     R1,     $SP     ;SAVE "STACK POINTER"
  (1)  007572  122767  000200  170632           CMPB    #200,   $PSW    ;CHECK PS (EXCEPT T BIT)
  (1)  007600  001402                           BEQ     .+6             ;BRANCH IF OK
  (1)  007602  104000                           HLT                     ;PS NOT EQUAL TO 200
  (3)  007604  000305                           305                     ;THE ERROR NUMBER IS 305
  (1)
  (1)  007606  022767  000514  170620           CMP     #STACK4,$SP     ;CHECK THE STACK POINTER (R1)
  (1)  007614  001402                           BEQ     .+6             ;BRANCH IF OK
  (1)  007616  104000                           HLT                     ;STACK POINTER (R1) NOT EQUAL TO #STACK4
  (3)  007620  000306                           306                     ;THE ERROR NUMBER IS 306
  (1)
  (1)  007622  022767  000200  170606           CMP     #000200,ANS1    ;CHECK FIRST HALF OF ANSWER
  (1)  007630  001402                           BEQ     .+6             ;BRANCH IF OK
  (1)  007632  104002                           HLT+2                   ;ANS1 NOT EQUAL TO 000200
  (3)  007634  000307                           307                     ;THE ERROR NUMBER IS 307
  (1)
  (1)  007636  005767  170576                   TST     ANS2            ;CHECK SECOND HALF OF ANSWER
  (1)  007642  001402                           BEQ     .+6             ;BRANCH IF OK
  (1)  007644  104002                           HLT+2                   ;ANS2 NOT EQUAL TO 000000
  (3)  007646  000310                           310                     ;THE ERROR NUMBER IS 310
  (1)
  (1)  007650  122767  000042  170526  END42:   CMPB    #42,    $TESTN  ;CHECK THE TEST NUMBER
  (1)  007656  001402                           BEQ     .+6             ;BRANCH IF OK
  (1)  007660  104000                           HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
  (3)  007662  000311                           311                     ;THE ERROR NUMBER IS 311
  (1)
  (1)
```

```
 7200
  (1)                                ;*************************************************************************
  (1)                                ;TEST 43:       FSUB (LSI-11 FLOATING SUBTRACT INSTRUCTION)
  (1)                                ;       104000,105004 - 104000,104000 = 100401,000000
  (1)                                ;       PS = 210,       STACK POINTER = PC
  (1)                                ;*************************************************************************
  (1)
  (1)  007664  104400                          SCOPE
  (1)  007666  004567  005644        TST43:    JSR     R5,     PUSH7   ;PUSH 4 WORDS ONTO STACK, SET PRIORITY
  (1)  007672  007716                          .WORD   STK43           ;TOP OF STACK
  (1)  007674  104000  104000                  .WORD   104000,104000   ;SECOND OPERAND ON TOP
  (1)  007700  104000  105004                  .WORD   104000,105004   ;FIRST OPERAND ON BOTTOM
  (1)  007704  000252                          .WORD   252             ;PROCESSOR PRIORITY LEVEL
  (1)  007706  015634  000340                  .WORD   TRAPER,340      ;FIS TRAP VECTOR
  (1)
  (1)  007712  000240                          NOP
  (1)  007714  075017                          FSUB    PC              ;FLOATING SUBTRACT ON FOLLOWING 4 WORDS
  (1)  007716  104000                STK43:    104000                  ;SHOULD CONTAIN 104000
  (1)  007720  104000                          104000                  ;SHOULD CONTAIN 104000
  (1)  007722  104000                          104000                  ;BEFORE FSUB, 104000; AFTER, 100401
  (1)  007724  105004                          105004                  ;BEFORE FSUB, 105004; AFTER, 000000
  (1)
  (1)  007726  004767  005634                  JSR     PC,     POP7    ;POP THE ANSWER
  (1)  007732  122767  000210  170472          CMPB    #210,   $PSW    ;CHECK PS (EXCEPT T BIT)
  (1)  007740  001402                          BEQ     .+6             ;BRANCH IF OK
  (1)  007742  104000                          HLT                     ;PS NOT EQUAL TO 210
  (3)  007744  000312                          312                     ;THE ERROR NUMBER IS 312
  (1)
  (1)  007746  022767  104000  170462          CMP     #104000,ANS1    ;CHECK FIRST HALF OF INPUT DATA (STK43)
  (1)  007754  001402                          BEQ     .+6             ;BRANCH IF OK
  (1)  007756  104002                          HLT+2                   ;ANS1 NOT EQUAL TO 104000
  (3)  007760  000313                          313                     ;THE ERROR NUMBER IS 313
  (1)
  (1)  007762  022767  104000  170450          CMP     #104000,ANS2    ;CHECK SECOND HALF OF INPUT DATA (STK43+2)
  (1)  007770  001402                          BEQ     .+6             ;BRANCH IF OK
  (1)  007772  104002                          HLT+2                   ;ANS2 NOT EQUAL TO 104000
  (3)  007774  000314                          314                     ;THE ERROR NUMBER IS 314
  (1)
  (1)  007776  022767  100401  170436          CMP     #100401,ANS3    ;CHECK FIRST HALF OF ANSWER
  (1)  010004  001402                          BEQ     .+6             ;BRANCH IF OK
  (1)  010006  104004                          HLT+4                   ;ANS3 NOT EQUAL TO 100401
  (3)  010010  000315                          315                     ;THE ERROR NUMBER IS 315
  (1)
  (1)  010012  005767  170426                  TST     ANS4            ;CHECK SECOND HALF OF ANSWER
  (1)  010016  001402                          BEQ     .+6             ;BRANCH IF OK
  (1)  010020  104004                          HLT+4                   ;ANS4 NOT EQUAL TO 000000
  (3)  010022  000316                          316                     ;THE ERROR NUMBER IS 316
  (1)
  (1)  010024  122767  000043  170352  END43:  CMPB    #43,    $TESTN  ;CHECK THE TEST NUMBER
  (1)  010032  001402                          BEQ     .+6             ;BRANCH IF OK
  (1)  010034  104000                          HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
  (3)  010036  000317                          317                     ;THE ERROR NUMBER IS 317
  (1)
  (1)
```

```
7202
 (1)                                ;*************************************************************************
 (1)                                ;TEST 44:       FMUL (LSI-11 FLOATING MULTIPLY INSTRUCTION)
 (1)                                ;       134600,073601 * 104000,104000 = 000401,000000
 (1)                                ;       PS = 200,       STACK POINTER = PC
 (1)                                ;*************************************************************************
 (1)
 (1)  010040  104400                            SCOPE
 (1)  010042  004567  005470        TST44:      JSR     R5,     PUSH7   ;PUSH 4 WORDS ONTO STACK, SET PRIORITY
 (1)  010046  010072                            .WORD   STK44           ;TOP OF STACK
 (1)  010050  104000  104000                    .WORD   104000,104000   ;SECOND OPERAND ON TOP
 (1)  010054  134600  073601                    .WORD   134600,073601   ;FIRST OPERAND ON BOTTOM
 (1)  010060  000246                            .WORD   246             ;PROCESSOR PRIORITY LEVEL
 (1)  010062  015634  000340                    .WORD   TRAPER,340      ;FIS TRAP VECTOR
 (1)
 (1)  010066  000240                            NOP
 (1)  010070  075027                            FMUL    PC              ;FLOATING MULTIPLY ON FOLLOWING 4 WORDS
 (1)  010072  104000                STK44:      104000                  ;SHOULD CONTAIN 104000
 (1)  010074  104000                            104000                  ;SHOULD CONTAIN 104000
 (1)  010076  134600                            134600                  ;BEFORE FMUL, 134600; AFTER, 000401
 (1)  010100  073601                            073601                  ;BEFORE FMUL, 073601; AFTER, 000000
 (1)
 (1)  010102  004767  005460                    JSR     PC,     POP7    ;POP THE ANSWER
 (1)  010106  122767  000200  170316            CMPB    #200,   $PSW    ;CHECK PS (EXCEPT T BIT)
 (1)  010114  001402                            BEQ     .+6             ;BRANCH IF OK
 (1)  010116  104000                            HLT                     ;PS NOT EQUAL TO 200
 (3)  010120  000320                            320                     ;THE ERROR NUMBER IS 320
 (1)
 (1)  010122  022767  104000  170306            CMP     #104000,ANS1    ;CHECK FIRST HALF OF INPUT DATA (STK44)
 (1)  010130  001402                            BEQ     .+6             ;BRANCH IF OK
 (1)  010132  104002                            HLT+2                   ;ANS1 NOT EQUAL TO 104000
 (3)  010134  000321                            321                     ;THE ERROR NUMBER IS 321
 (1)
 (1)  010136  022767  104000  170274            CMP     #104000,ANS2    ;CHECK SECOND HALF OF INPUT DATA (STK44+2)
 (1)  010144  001402                            BEQ     .+6             ;BRANCH IF OK
 (1)  010146  104002                            HLT+2                   ;ANS2 NOT EQUAL TO 104000
 (3)  010150  000322                            322                     ;THE ERROR NUMBER IS 322
 (1)
 (1)  010152  022767  000401  170262            CMP     #000401,ANS3    ;CHECK FIRST HALF OF ANSWER
 (1)  010160  001402                            BEQ     .+6             ;BRANCH IF OK
 (1)  010162  104004                            HLT+4                   ;ANS3 NOT EQUAL TO 000401
 (3)  010164  000323                            323                     ;THE ERROR NUMBER IS 323
 (1)
 (1)  010166  005767  170252                    TST     ANS4            ;CHECK SECOND HALF OF ANSWER
 (1)  010172  001402                            BEQ     .+6             ;BRANCH IF OK
 (1)  010174  104004                            HLT+4                   ;ANS4 NOT EQUAL TO 000000
 (3)  010176  000324                            324                     ;THE ERROR NUMBER IS 324
 (1)
 (1)  010200  122767  000044  170176  END44:     CMPB    #44,    $TESTN  ;CHECK THE TEST NUMBER
 (1)  010206  001402                            BEQ     .+6             ;BRANCH IF OK
 (1)  010210  104000                            HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
 (3)  010212  000325                            325                     ;THE ERROR NUMBER IS 325
 (1)
 (1)
7206
 (1)                                ;*************************************************************************
```

```
(1)                                     ;TEST 45:       FMUL (LSI-11 FLOATING MULTIPLY INSTRUCTION)
(1)                                     ;       024252,125252 * 114100,000000 ==> UNDERFLOW
(1)                                     ;       PS(ON STACK) = 212,     STACK POINTER = R0
(1)                                     ;*************************************************************************
(1)
(1)  010214  104400                             SCOPE
(1)  010216  004567  005142             TST45:  JSR     R5,     PUSHR   ;PUSH 4 WORDS ONTO R0 STACK, SET PRIORITY
(1)  010222  114100  000000                     .WORD   114100,000000   ;SECOND OPERAND ON TOP
(1)  010226  024252  125252                     .WORD   024252,125252   ;FIRST OPERAND ON BOTTOM
(1)  010232  000305                             .WORD   305             ;PROCESSOR PRIORITY LEVEL
(1)  010234  010266  000057                     .WORD   ISR45,  057     ;FIS TRAP VECTOR
(1)  010240  012700  000510                     MOV     #STACK0,R0      ;SET UP R0 AS STACK POINTER
(1)
(1)  010244  000240                             NOP
(1)  010246  075020                             FMUL    R0              ;FLOATING MULTIPLY ON THE R0 STACK
(1)
(1)  010250  004767  005142             RTA45:  JSR     PC,     POPR    ;POP THE "ANSWER"
(1)  010254  010067  170154                     MOV     R0,     $SP     ;SAVE STACK POINTER (R0)
(1)  010260  104002                             HLT+2                   ;FIS TRAP DIDN'T OCCURE!
(3)  010262  000326                             326                     ;THE ERROR NUMBER IS 326
(1)  010264  000463                             BR      END45
(1)
(1)  010266  004767  005154             ISR45:  JSR     PC,     POPER   ;POP ALL DATA OFF THE STACKS
(1)  010272  010067  170136                     MOV     R0,     $SP     ;SAVE STACK POINTER (R0)
(1)  010276  122767  000057  170126             CMPB    #057,   $PSW    ;CHECK PS AFTER FIS TRAP
(1)  010304  001402                             BEQ     .+6             ;BRANCH IF OK
(1)  010306  104000                             HLT                     ;PS AFTER FIS TRAP NOT EQUAL TO 057
(3)  010310  000327                             327                     ;THE ERROR NUMBER IS 327
(1)
(1)  010312  022767  000510  170114             CMP     #STACK0,$SP     ;CHECK THE STACK POINTER (R0)
(1)  010320  001402                             BEQ     .+6             ;BRANCH IF OK
(1)  010322  104000                             HLT                     ;STACK POINTER (R0) NOT EQUAL TO #STACK0
(3)  010324  000330                             330                     ;THE ERROR NUMBER IS 330
(1)
(1)  010326  022767  010250  170102             CMP     #RTA45, ANS1    ;CHECK FIS TRAP RETURN ADDRESS
(1)  010334  001402                             BEQ     .+6             ;BRANCH IF OK
(1)  010336  104001                             HLT+1                   ;FIS TRAP AT WRONG ADDRESS
(3)  010340  000331                             331                     ;THE ERROR NUMBER IS 331
(1)
(1)  010342  022767  000212  170070             CMP     #212,   ANS2    ;CHECK PS BEFORE FIS TRAP
(1)  010350  001402                             BEQ     .+6             ;BRANCH IF OK
(1)  010352  104002                             HLT+2                   ;PS AT FIS TRAP TIME NOT 212
(3)  010354  000332                             332                     ;THE ERROR NUMBER IS 332
(1)
(1)  010356  022767  114100  170056             CMP     #114100,ANS3    ;CHECK DATA FROM THE STACK
(1)  010364  001402                             BEQ     .+6             ;BRANCH IF OK
(1)  010366  104004                             HLT+4                   ;DATA ON STACK (114100) CHANGED
(3)  010370  000333                             333                     ;THE ERROR NUMBER IS 333
(1)
(1)  010372  005767  170046                     TST     ANS4            ;CHECK DATA FROM STACK
(1)  010376  001402                             BEQ     .+6             ;BRANCH IF OK
(1)  010400  104004                             HLT+4                   ;DATA ON STACK (000000) CHANGED
(3)  010402  000334                             334                     ;THE ERROR NUMBER IS 334
(1)
(1)  010404  022767  024252  170034             CMP     #024252,ANS5    ;CHECK DATA FROM STACK
(1)  010412  001402                             BEQ     .+6             ;BRANCH IF OK
```

```
(1)  010414  104006                          HLT+6                    ;DATA ON STACK (024252) CHANGED
(3)  010416  000335                          335                      ;THE ERROR NUMBER IS 335
(1)
(1)  010420  022767  125252  170022          CMP     #125252,ANS6     ;CHECK DATA FROM STACK
(1)  010426  001402                          BEQ     .+6              ;BRANCH IF OK
(1)  010430  104006                          HLT+6                    ;DATA ON STACK (125252) CHANGED
(3)  010432  000336                          336                      ;THE ERROR NUMBER IS 336
(1)
(1)  010434  122767  000045  167742  END45:  CMPB    #45,    $TESTN   ;CHECK THE TEST NUMBER
(1)  010442  001402                          BEQ     .+6              ;BRANCH IF OK
(1)  010444  104000                          HLT                      ;WRONG TEST!  PC MUST HAVE FOULED UP.
(3)  010446  000337                          337                      ;THE ERROR NUMBER IS 337
(1)
(1)
```

```
 7210
  (1)                           ;*****************************************************************************
  (1)                           ;TEST 46:       FMUL (LSI-11 FLOATING MULTIPLY INSTRUCTION)
  (1)                           ;       076452,125252 * 041500,000001 ==> OVERFLOW
  (1)                           ;       PS(ON STACK) = 002,     STACK POINTER = SP
  (1)                           ;*****************************************************************************
  (1)
  (1)  010450  104400                           SCOPE
  (1)  010452  004567  004534   TST46:  JSR     R5      PUSHS   ;PUSH 4 WORDS ONTO STACK, SET PRIORITY
  (1)  010456  041500  000001           .WORD   041500,000001   ;SECOND OPERAND ON TOP
  (1)  010462  076452  125252           .WORD   076452,125252   ;FIRST OPERAND ON BOTTOM
  (1)  010466  000105                   .WORD   105             ;PROCESSOR PRIORITY LEVEL
  (1)  010470  010516  000357           .WORD   ISR46,  357     ;FIS TRAP VECTOR
  (1)
  (1)  010474  000240                   NOP
  (1)  010476  075026                   FMUL    SP              ;FLOATING MULTIPLY ON THE STACK
  (1)
  (1)  010500  004767  004546   RTA46:  JSR     PC,     POPS    ;POP THE "ANSWER"
  (1)  010504  104002                   HLT+2                   ;FIS TRAP DIDN'T OCCURE!
  (3)  010506  000340                   340                     ;THE ERROR NUMBER IS 340
  (1)  010510  012706  000600           MOV     #BEGIN, SP      ;RESTORE THE STACK POINTER
  (1)  010514  000464                   BR      END46
  (1)
  (1)  010516  004767  004562   ISR46:  JSR     PC,     POPES   ;POP ALL DATA OFF THE STACK
  (1)  010522  022706  000600           CMP     #BEGIN, SP      ;CHECK THE STACK POINTER
  (1)  010526  001405                   BEQ     ISA46           ;BRANCH IF OK
  (1)  010530  012706  000600           MOV     #BEGIN, SP      ;RESTORE THE STACK POINTER
  (1)  010534  104000                   HLT                     ;STACK POINTER FOULED UP
  (3)  010536  000341                   341                     ;THE ERROR NUMBER IS 341
  (1)  010540  000452                   BR      END46           ;SKIP REST OF TEST
  (1)
  (1)  010542  122767  000357  167662  ISA46:  CMPB    #357,   $PSW    ;CHECK PS AFTER FIS TRAP
  (1)  010550  001402                   BEQ     .+6             ;BRANCH IF OK
  (1)  010552  104000                   HLT                     ;PS AFTER FIS TRAP NOT EQUAL TO 357
  (3)  010554  000342                   342                     ;THE ERROR NUMBER IS 342
  (1)
  (1)  010556  022767  010500  167652           CMP     #RTA46, ANS1    ;CHECK FIS TRAP RETURN ADDRESS
  (1)  010564  001402                   BEQ     .+6             ;BRANCH IF OK
  (1)  010566  104001                   HLT+1                   ;FIS TRAP AT WRONG ADDRESS
  (3)  010570  000343                   343                     ;THE ERROR NUMBER IS 343
  (1)
  (1)  010572  022767  000002  167640           CMP     #002,   ANS2    ;CHECK PS BEFORE FIS TRAP
  (1)  010600  001402                   BEQ     .+6             ;BRANCH IF OK
  (1)  010602  104002                   HLT+2                   ;PS AT FIS TRAP TIME NOT 002
  (3)  010604  000344                   344                     ;THE ERROR NUMBER IS 344
  (1)
  (1)  010606  022767  041500  167626           CMP     #041500,ANS3    ;CHECK DATA FROM THE STACK
  (1)  010614  001402                   BEQ     .+6             ;BRANCH IF OK
  (1)  010616  104004                   HLT+4                   ;DATA ON STACK (041500) CHANGED
  (3)  010620  000345                   345                     ;THE ERROR NUMBER IS 345
  (1)
  (1)  010622  022767  000001  167614           CMP     #000001,ANS4    ;CHECK DATA FROM STACK
  (1)  010630  001402                   BEQ     .+6             ;BRANCH IF OK
  (1)  010632  104004                   HLT+4                   ;DATA ON STACK (000001) CHANGED
  (3)  010634  000346                   346                     ;THE ERROR NUMBER IS 346
  (1)
```

```
(1)  010636  022767  076452  167602          CMP     #076452,ANS5    ;CHECK DATA FROM STACK
(1)  010644  001402                          BEQ     .+6             ;BRANCH IF OK
(1)  010646  104006                          HLT+6                   ;DATA ON STACK (076452) CHANGED
(3)  010650  000347                          347                     ;THE ERROR NUMBER IS 347
(1)
(1)  010652  022767  125252  167570          CMP     #125252,ANS6    ;CHECK DATA FROM STACK
(1)  010660  001402                          BEQ     .+6             ;BRANCH IF OK
(1)  010662  104006                          HLT+6                   ;DATA ON STACK (125252) CHANGED
(3)  010664  000350                          350                     ;THE ERROR NUMBER IS 350
(1)
(1)  010666  122767  000046  167510  END46:  CMPB    #46,    $TESTN  ;CHECK THE TEST NUMBER
(1)  010674  001402                          BEQ     .+6             ;BRANCH IF OK
(1)  010676  104000                          HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
(3)  010700  000351                          351                     ;THE ERROR NUMBER IS 351
(1)
(1)
```

```
7214
 (1)
 (1)                                ;*****************************************************************************
 (1)                                ;TEST 47:       FDIV (LSI-11 FLOATING DIVIDE INSTRUCTION)
 (1)                                ;       167452,125251 / 127652,125252 = 077777,177776
 (1)                                ;       PS = 000,       STACK POINTER = R0
 (1)                                ;*****************************************************************************
 (1)
 (1)  010702  104400                            SCOPE
 (1)  010704  004567  004454        TST47:      JSR     R5,     PUSHR   ;PUSH 4 WORDS ONTO R0 STACK, SET PRIORITY
 (1)  010710  127652  125252                    .WORD   127652,125252   ;SECOND OPERAND ON TOP
 (1)  010714  167452  125251                    .WORD   167452,125251   ;FIRST OPERAND ON BOTTOM
 (1)  010720  000111                            .WORD   111             ;PROCESSOR PRIORITY LEVEL
 (1)  010722  015634  000340                    .WORD   TRAPER, 340     ;FIS TRAP VECTOR
 (1)  010726  012700  000510                    MOV     #STACK0,R0      ;SHECK  STACK POINTER
 (1)
 (1)  010732  000240                            NOP
 (1)  010734  075030                            FDIV    R0              ;FLOATING DIVIDE ON THE R0 STACK
 (1)
 (1)  010736  004767  004454                    JSR     PC,     POPR    ;POP THE ANSWER
 (1)  010742  010067  167466                    MOV     R0,     $SP     ;SAVE "STACK POINTER"
 (1)  010746  105767  167460                    TSTB    $PSW            ;CHECK PS (EXCEPT T BIT)
 (1)  010752  001402                            BEQ     .+6             ;BRANCH IF OK
 (1)  010754  104000                            HLT                     ;PS NOT EQUAL TO 000
 (3)  010756  000352                            352                     ;THE ERROR NUMBER IS 352
 (1)
 (1)  010760  022767  000514  167446            CMP     #STACK4,$SP     ;CHECK THE STACK POINTER (R0)
 (1)  010766  001402                            BEQ     .+6             ;BRANCH IF OK
 (1)  010770  104000                            HLT                     ;STACK POINTER (R0) NOT EQUAL TO #STACK4
 (3)  010772  000353                            353                     ;THE ERROR NUMBER IS 353
 (1)
 (1)  010774  022767  077777  167434            CMP     #077777,ANS1    ;CHECK FIRST HALF OF ANSWER
 (1)  011002  001402                            BEQ     .+6             ;BRANCH IF OK
 (1)  011004  104002                            HLT+2                   ;ANS1 NOT EQUAL TO 077777
 (3)  011006  000354                            354                     ;THE ERROR NUMBER IS 354
 (1)
 (1)  011010  022767  177776  167422            CMP     #177776,ANS2    ;CHECK SECOND HALF OF ANSWER
 (1)  011016  001402                            BEQ     .+6             ;BRANCH IF OK
 (1)  011020  104002                            HLT+2                   ;ANS2 NOT EQUAL TO 177776
 (3)  011022  000355                            355                     ;THE ERROR NUMBER IS 355
 (1)
 (1)  011024  122767  000047  167352  END47:    CMPB    #47,    $TESTN  ;CHECK THE TEST NUMBER
 (1)  011032  001402                            BEQ     .+6             ;BRANCH IF OK
 (1)  011034  104000                            HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
 (3)  011036  000356                            356                     ;THE ERROR NUMBER IS 356
 (1)
 (1)
```

```
 7215
  (1)                                ;*****************************************************************************
  (1)                                ;TEST 50:       FDIV (LSI-11 FLOATING DIVIDE INSTRUCTION)
  (1)                                ;       167452,125252 / 027652,125253 = 177777,177777
  (1)                                ;       PS = 210,       STACK POINTER = SP
  (1)                                ;*****************************************************************************
  (1)
  (1)  011040  104400                           SCOPE
  (1)  011042  004567  004144        TST50:     JSR     R5,     PUSHS   ;PUSH 4 WORDS ONTO STACK, SET PRIORITY
  (1)  011046  027652  125253                   .WORD   027652,125253   ;SECOND OPERAND ON TOP
  (1)  011052  167452  125252                   .WORD   167452,125252   ;FIRST OPERAND ON BOTTOM
  (1)  011056  000300                           .WORD   300             ;PROCESSOR PRIORITY LEVEL
  (1)  011060  015634  000340                   .WORD   TRAPER, 340     ;FIS TRAP VECTOR
  (1)
  (1)  011064  000240                           NOP
  (1)  011066  075036                           FDIV    SP              ;FLOATING DIVIDE ON THE STACK
  (1)
  (1)  011070  004767  004156                   JSR     PC,     POPS    ;POP THE ANSWER
  (1)  011074  022706  000600                   CMP     #BEGIN, SP      ;CHECK THE STACK POINTER
  (1)  011100  001405                           BEQ     TSA50           ;BRANCH IF OK
  (1)  011102  012706  000600                   MOV     #BEGIN, SP      ;RESTORE STACK POINTER
  (1)  011106  104000                           HLT                     ;STACK POINTER FOULED UP
  (3)  011110  000357                           357                     ;THE ERROR NUMBER IS 357
  (1)  011112  000422                           BR      ENDS0           ;SKIP REST OF TEST
  (1)
  (1)  011114  122767  000210  167310  TSA50:   CMPB    #210,   $PSW    ;CHECK PS (EXCEPT T BIT)
  (1)  011122  001402                           BEQ     .+6             ;BRANCH IF OK
  (1)  011124  104000                           HLT                     ;PS NOT EQUAL TO 210
  (3)  011126  000360                           360                     ;THE ERROR NUMBER IS 360
  (1)
  (1)  011130  022767  177777  167300          CMP     #177777,ANS1    ;CHECK FIRST HALF OF ANSWER
  (1)  011136  001402                           BEQ     .+6             ;BRANCH IF OK
  (1)  011140  104002                           HLT+2                   ;ANS1 NOT EQUAL TO 177777
  (3)  011142  000361                           361                     ;THE ERROR NUMBER IS 361
  (1)
  (1)  011144  022767  177777  167266          CMP     #177777,ANS2    ;CHECK SECOND HALF OF ANSWER
  (1)  011152  001402                           BEQ     .+6             ;BRANCH IF OK
  (1)  011154  104002                           HLT+2                   ;ANS2 NOT EQUAL TO 177777
  (3)  011156  000362                           362                     ;THE ERROR NUMBER IS 362
  (1)
  (1)  011160  122767  000050  167216  ENDS0:   CMPB    #50,    $TESTN  ;CHECK THE TEST NUMBER
  (1)  011166  001402                           BEQ     .+6             ;BRANCH IF OK
  (1)  011170  104000                           HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
  (3)  011172  000363                           363                     ;THE ERROR NUMBER IS 363
  (1)
  (1)
```

```
 7216
  (1)
  (1)                                   ;*****************************************************************************
  (1)                                   ;TEST 51:       FDIV (LSI-11 FLOATING DIVIDE INSTRUCTION)
  (1)                                   ;       125252,125252 / 065252,125252 = 100200,000000
  (1)                                   ;       PS = 210,       STACK POINTER = R2
  (1)                                   ;*****************************************************************************
  (1)
  (1)  011174  104400                           SCOPE
  (1)  011176  004567  004162           TST51:  JSR     R5,     PUSHR   ;PUSH 4 WORDS ONTO R2 STACK, SET PRIORITY
  (1)  011202  065252  125252                   .WORD   065252,125252   ;SECOND OPERAND ON TOP
  (1)  011206  125252  125252                   .WORD   125252,125252   ;FIRST OPERAND ON BOTTOM
  (1)  011212  000217                           .WORD   217             ;PROCESSOR PRIORITY LEVEL
  (1)  011214  015634  000340                   .WORD   TRAPER, 340     ;FIS TRAP VECTOR
  (1)  011220  012702  000510                   MOV     #STACK0,R2      ;SHECK  STACK POINTER
  (1)
  (1)  011224  000240                           NOP
  (1)  011226  075032                           FDIV    R2              ;FLOATING DIVIDE ON THE R2 STACK
  (1)
  (1)  011230  004767  004162                   JSR     PC,     POPR    ;POP THE ANSWER
  (1)  011234  010267  167174                   MOV     R2,     $SP     ;SAVE "STACK POINTER"
  (1)  011240  122767  000210  167164           CMPB    #210,   $PSW    ;CHECK PS (EXCEPT T BIT)
  (1)  011246  001402                           BEQ     .+6             ;BRANCH IF OK
  (1)  011250  104000                           HLT                     ;PS NOT EQUAL TO 210
  (3)  011252  000364                           364                     ;THE ERROR NUMBER IS 364
  (1)
  (1)  011254  022767  000514  167152           CMP     #STACK4,$SP     ;CHECK THE STACK POINTER (R2)
  (1)  011262  001402                           BEQ     .+6             ;BRANCH IF OK
  (1)  011264  104000                           HLT                     ;STACK POINTER (R2) NOT EQUAL TO #STACK4
  (3)  011266  000365                           365                     ;THE ERROR NUMBER IS 365
  (1)
  (1)  011270  022767  100200  167140           CMP     #100200,ANS1    ;CHECK FIRST HALF OF ANSWER
  (1)  011276  001402                           BEQ     .+6             ;BRANCH IF OK
  (1)  011300  104002                           HLT+2                   ;ANS1 NOT EQUAL TO 100200
  (3)  011302  000366                           366                     ;THE ERROR NUMBER IS 366
  (1)
  (1)  011304  005767  167130                   TST     ANS2            ;CHECK SECOND HALF OF ANSWER
  (1)  011310  001402                           BEQ     .+6             ;BRANCH IF OK
  (1)  011312  104002                           HLT+2                   ;ANS2 NOT EQUAL TO 000000
  (3)  011314  000367                           367                     ;THE ERROR NUMBER IS 367
  (1)
  (1)  011316  122767  000051  167060  END51:  CMPB    #51,    $TESTN  ;CHECK THE TEST NUMBER
  (1)  011324  001402                           BEQ     .+6             ;BRANCH IF OK
  (1)  011326  104000                           HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
  (3)  011330  000370                           370                     ;THE ERROR NUMBER IS 370
  (1)
  (1)
```

```
 7217
  (1)                                ;*****************************************************************************
  (1)                                ;TEST 52:       FDIV (LSI-11 FLOATING DIVIDE INSTRUCTION)
  (1)                                ;       000000,000000 / 140670,123456 = 000000,000000
  (1)                                ;       PS = 004,       STACK POINTER = R3
  (1)                                ;*****************************************************************************
  (1)
  (1)  011332  104400                           SCOPE
  (1)  011334  004567  004024        TST52:     JSR     R5,     PUSHR   ;PUSH 4 WORDS ONTO R3 STACK, SET PRIORITY
  (1)  011340  140670  123456                   .WORD   140670,123456   ;SECOND OPERAND ON TOP
  (1)  011344  000000  000000                   .WORD   000000,000000   ;FIRST OPERAND ON BOTTOM
  (1)  011350  000105                           .WORD   105             ;PROCESSOR PRIORITY LEVEL
  (1)  011352  015634  000340                   .WORD   TRAPER, 340     ;FIS TRAP VECTOR
  (1)  011356  012703  000510                   MOV     #STACK0,R3      ;SHECK  STACK POINTER
  (1)
  (1)  011362  000240                           NOP
  (1)  011364  075033                           FDIV    R3              ;FLOATING DIVIDE ON THE R3 STACK
  (1)
  (1)  011366  004767  004024                   JSR     PC,     POPR    ;POP THE ANSWER
  (1)  011372  010367  167036                   MOV     R3,     $SP     ;SAVE "STACK POINTER"
  (1)  011376  122767  000004  167026           CMPB    #004,   $PSW    ;CHECK PS (EXCEPT T BIT)
  (1)  011404  001402                           BEQ     .+6             ;BRANCH IF OK
  (1)  011406  104000                           HLT                     ;PS NOT EQUAL TO 004
  (3)  011410  000371                           371                     ;THE ERROR NUMBER IS 371
  (1)
  (1)  011412  022767  000514  167014           CMP     #STACK4,$SP     ;CHECK THE STACK POINTER (R3)
  (1)  011420  001402                           BEQ     .+6             ;BRANCH IF OK
  (1)  011422  104000                           HLT                     ;STACK POINTER (R3) NOT EQUAL TO #STACK4
  (3)  011424  000372                           372                     ;THE ERROR NUMBER IS 372
  (1)
  (1)  011426  005767  167004                   TST     ANS1            ;CHECK FIRST HALF OF ANSWER
  (1)  011432  001402                           BEQ     .+6             ;BRANCH IF OK
  (1)  011434  104002                           HLT+2                   ;ANS1 NOT EQUAL TO 000000
  (3)  011436  000373                           373                     ;THE ERROR NUMBER IS 373
  (1)
  (1)  011440  005767  166774                   TST     ANS2            ;CHECK SECOND HALF OF ANSWER
  (1)  011444  001402                           BEQ     .+6             ;BRANCH IF OK
  (1)  011446  104002                           HLT+2                   ;ANS2 NOT EQUAL TO 000000
  (3)  011450  000374                           374                     ;THE ERROR NUMBER IS 374
  (1)
  (1)  011452  122767  000052  166724  END52:   CMPB    #52,    $TESTN  ;CHECK THE TEST NUMBER
  (1)  011460  001402                           BEQ     .+6             ;BRANCH IF OK
  (1)  011462  104000                           HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
  (3)  011464  000375                           375                     ;THE ERROR NUMBER IS 375
  (1)
  (1)
```

```
7219
 (1)                                ;*****************************************************************************
 (1)                                ;TEST 53:       FDIV (LSI-11 FLOATING DIVIDE INSTRUCTION)
 (1)                                ;       102500,146000 / 104000,104000 = 036700,000000
 (1)                                ;       PS = 200,       STACK POINTER = PC
 (1)                                ;*****************************************************************************
 (1)
 (1)  011466  104400                            SCOPE
 (1)  011470  004567  004042        TST53:      JSR     R5,     PUSH7   ;PUSH 4 WORDS ONTO STACK, SET PRIORITY
 (1)  011474  011520                            .WORD   STK53           ;TOP OF STACK
 (1)  011476  104000  104000                    .WORD   104000,104000   ;SECOND OPERAND ON TOP
 (1)  011502  102500  146000                    .WORD   102500,146000   ;FIRST OPERAND ON BOTTOM
 (1)  011506  000357                            .WORD   357             ;PROCESSOR PRIORITY LEVEL
 (1)  011510  015634  000340                    .WORD   TRAPER,340      ;FIS TRAP VECTOR
 (1)
 (1)  011514  000240                            NOP
 (1)  011516  075037                            FDIV    PC              ;FLOATING DIVIDE ON FOLLOWING 4 WORDS
 (1)  011520  104000                STK53:      104000                  ;SHOULD CONTAIN 104000
 (1)  011522  104000                            104000                  ;SHOULD CONTAIN 104000
 (1)  011524  102500                            102500                  ;BEFORE FDIV, 102500; AFTER, 036700
 (1)  011526  146000                            146000                  ;BEFORE FDIV, 146000; AFTER, 000000
 (1)
 (1)  011530  004767  004032                    JSR     PC,     POP7    ;POP THE ANSWER
 (1)  011534  122767  000200  166670            CMPB    #200,   $PSW    ;CHECK PS (EXCEPT T BIT)
 (1)  011542  001402                            BEQ     .+6             ;BRANCH IF OK
 (1)  011544  104000                            HLT                     ;PS NOT EQUAL TO 200
 (3)  011546  000376                            376                     ;THE ERROR NUMBER IS 376
 (1)
 (1)  011550  022767  104000  166660            CMP     #104000,ANS1    ;CHECK FIRST HALF OF INPUT DATA (STK53)
 (1)  011556  001402                            BEQ     .+6             ;BRANCH IF OK
 (1)  011560  104002                            HLT+2                   ;ANS1 NOT EQUAL TO 104000
 (3)  011562  000377                            377                     ;THE ERROR NUMBER IS 377
 (1)
 (1)  011564  022767  104000  166646            CMP     #104000,ANS2    ;CHECK SECOND HALF OF INPUT DATA (STK53+2)
 (1)  011572  001402                            BEQ     .+6             ;BRANCH IF OK
 (1)  011574  104002                            HLT+2                   ;ANS2 NOT EQUAL TO 104000
 (3)  011576  000400                            400                     ;THE ERROR NUMBER IS 400
 (1)
 (1)  011600  022767  036700  166634            CMP     #036700,ANS3    ;CHECK FIRST HALF OF ANSWER
 (1)  011606  001402                            BEQ     .+6             ;BRANCH IF OK
 (1)  011610  104004                            HLT+4                   ;ANS3 NOT EQUAL TO 036700
 (3)  011612  000401                            401                     ;THE ERROR NUMBER IS 401
 (1)
 (1)  011614  005767  166624                    TST     ANS4            ;CHECK SECOND HALF OF ANSWER
 (1)  011620  001402                            BEQ     .+6             ;BRANCH IF OK
 (1)  011622  104004                            HLT+4                   ;ANS4 NOT EQUAL TO 000000
 (3)  011624  000402                            402                     ;THE ERROR NUMBER IS 402
 (1)
 (1)  011626  122767  000053  166550  END53:     CMPB    #53,    $TESTN  ;CHECK THE TEST NUMBER
 (1)  011634  001402                            BEQ     .+6             ;BRANCH IF OK
 (1)  011636  104000                            HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
 (3)  011640  000403                            403                     ;THE ERROR NUMBER IS 403
 (1)
 (1)
7223
 (1)                                ;*****************************************************************************
```

```
(1)                                     ;TEST 54:       FDIV (LSI-11 FLOATING DIVIDE INSTRUCTION)
(1)                                     ;               025252,125251 / 065252,125252 ==> UNDERFLOW
(1)                                     ;               PS(ON STACK) = 012,     STACK POINTER = R1
(1)                                     ;*********************************************************************************
(1)
(1)  011642  104400                             SCOPE
(1)  011644  004567  003514             TST54:  JSR     R5,     PUSHR   ;PUSH 4 WORDS ONTO R1 STACK, SET PRIORITY
(1)  011650  065252  125252                     .WORD   065252,125252   ;SECOND OPERAND ON TOP
(1)  011654  025252  125251                     .WORD   025252,125251   ;FIRST OPERAND ON BOTTOM
(1)  011660  000015                             .WORD   015             ;PROCESSOR PRIORITY LEVEL
(1)  011662  011714  000300                     .WORD   ISR54, 300      ;FIS TRAP VECTOR
(1)  011666  012701  000510                     MOV     #STACK0,R1      ;SET UP R1 AS STACK POINTER
(1)
(1)  011672  000240                             NOP
(1)  011674  075031                             FDIV    R1              ;FLOATING DIVIDE ON THE R1 STACK
(1)
(1)  011676  004767  003514             RTA54:  JSR     PC,     POPR    ;POP THE "ANSWER"
(1)  011702  010167  166526                     MOV     R1,     $SP     ;SAVE STACK POINTER (R1)
(1)  011706  104002                             HLT+2                   ;FIS TRAP DIDN'T OCCURE!
(3)  011710  000404                             404                     ;THE ERROR NUMBER IS 404
(1)  011712  000464                             BR      END54
(1)
(1)  011714  004767  003526             ISR54:  JSR     PC,     POPER   ;POP ALL DATA OFF THE STACKS
(1)  011720  010167  166510                     MOV     R1,     $SP     ;SAVE STACK POINTER (R1)
(1)  011724  122767  000300  166500             CMPB    #300,   $PSW    ;CHECK PS AFTER FIS TRAP
(1)  011732  001402                             BEQ     .+6             ;BRANCH IF OK
(1)  011734  104000                             HLT                     ;PS AFTER FIS TRAP NOT EQUAL TO 300
(3)  011736  000405                             405                     ;THE ERROR NUMBER IS 405
(1)
(1)  011740  022767  000510  166466             CMP     #STACK0,$SP     ;CHECK THE STACK POINTER (R1)
(1)  011746  001402                             BEQ     .+6             ;BRANCH IF OK
(1)  011750  104000                             HLT                     ;STACK POINTER (R1) NOT EQUAL TO #STACK0
(3)  011752  000406                             406                     ;THE ERROR NUMBER IS 406
(1)
(1)  011754  022767  011676  166454             CMP     #RTA54, ANS1    ;CHECK FIS TRAP RETURN ADDRESS
(1)  011762  001402                             BEQ     .+6             ;BRANCH IF OK
(1)  011764  104001                             HLT+1                   ;FIS TRAP AT WRONG ADDRESS
(3)  011766  000407                             407                     ;THE ERROR NUMBER IS 407
(1)
(1)  011770  022767  000012  166442             CMP     #012,   ANS2    ;CHECK PS BEFORE FIS TRAP
(1)  011776  001402                             BEQ     .+6             ;BRANCH IF OK
(1)  012000  104002                             HLT+2                   ;PS AT FIS TRAP TIME NOT 012
(3)  012002  000410                             410                     ;THE ERROR NUMBER IS 410
(1)
(1)  012004  022767  065252  166430             CMP     #065252,ANS3    ;CHECK DATA FROM THE STACK
(1)  012012  001402                             BEQ     .+6             ;BRANCH IF OK
(1)  012014  104004                             HLT+4                   ;DATA ON STACK (065252) CHANGED
(3)  012016  000411                             411                     ;THE ERROR NUMBER IS 411
(1)
(1)  012020  022767  125252  166416             CMP     #125252,ANS4    ;CHECK DATA FROM STACK
(1)  012026  001402                             BEQ     .+6             ;BRANCH IF OK
(1)  012030  104004                             HLT+4                   ;DATA ON STACK (125252) CHANGED
(3)  012032  000412                             412                     ;THE ERROR NUMBER IS 412
(1)
(1)  012034  022767  025252  166404             CMP     #025252,ANS5    ;CHECK DATA FROM STACK
(1)  012042  001402                             BEQ     .+6             ;BRANCH IF OK
```

```
   (1)  012044  104006                          HLT+6                   ;DATA ON STACK (025252) CHANGED
   (3)  012046  000413                          413                     ;THE ERROR NUMBER IS 413
   (1)
   (1)  012050  022767  125251  166372          CMP     #125251,ANS6    ;CHECK DATA FROM STACK
   (1)  012056  001402                          BEQ     .+6             ;BRANCH IF OK
   (1)  012060  104006                          HLT+6                   ;DATA ON STACK (125251) CHANGED
   (3)  012062  000414                          414                     ;THE ERROR NUMBER IS 414
   (1)
   (1)  012064  122767  000054  166312  END54:  CMPB    #54,    $TESTN  ;CHECK THE TEST NUMBER
   (1)  012072  001402                          BEQ     .+6             ;BRANCH IF OK
   (1)  012074  104000                          HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
   (3)  012076  000415                          415                     ;THE ERROR NUMBER IS 415
   (1)
   (1)
```

```
7227
 (1)                                ;*************************************************************************
 (1)                                ;TEST 55:       FDIV (LSI-11 FLOATING DIVIDE INSTRUCTION)
 (1)                                ;       067452,125252 / 127652,125252 ==> OVERFLOW
 (1)                                ;       PS(ON STACK) = 202,     STACK POINTER = R4
 (1)                                ;*************************************************************************
 (1)
 (1)  012100  104400                            SCOPE
 (1)  012102  004567  003256        TST55:  JSR     R5,     PUSHR   ;PUSH 4 WORDS ONTO R4 STACK, SET PRIORITY
 (1)  012106  127652  125252                .WORD   127652,125252   ;SECOND OPERAND ON TOP
 (1)  012112  067452  125252                .WORD   067452,125252   ;FIRST OPERAND ON BOTTOM
 (1)  012116  000242                        .WORD   242             ;PROCESSOR PRIORITY LEVEL
 (1)  012120  012152  000357                .WORD   ISR55, 357      ;FIS TRAP VECTOR
 (1)  012124  012704  000510                MOV     #STACK0,R4      ;SET UP R4 AS STACK POINTER
 (1)
 (1)  012130  000240                        NOP
 (1)  012132  075034                        FDIV    R4              ;FLOATING DIVIDE ON THE R4 STACK
 (1)
 (1)  012134  004767  003256        RTA55:  JSR     PC,     POPR    ;POP THE "ANSWER"
 (1)  012140  010467  166270                MOV     R4,     $SP     ;SAVE STACK POINTER (R4)
 (1)  012144  104002                        HLT+2                   ;FIS TRAP DIDN'T OCCURE!
 (3)  012146  000416                        416                     ;THE ERROR NUMBER IS 416
 (1)  012150  000464                        BR      END55
 (1)
 (1)  012152  004767  003270        ISR55:  JSR     PC,     POPER   ;POP ALL DATA OFF THE STACKS
 (1)  012156  010467  166252                MOV     R4,     $SP     ;SAVE STACK POINTER (R4)
 (1)  012162  122767  000357  166242        CMPB    #357,   $PSW    ;CHECK PS AFTER FIS TRAP
 (1)  012170  001402                        BEQ     .+6             ;BRANCH IF OK
 (1)  012172  104000                        HLT                     ;PS AFTER FIS TRAP NOT EQUAL TO 357
 (3)  012174  000417                        417                     ;THE ERROR NUMBER IS 417
 (1)
 (1)  012176  022767  000510  166230        CMP     #STACK0,$SP     ;CHECK THE STACK POINTER (R4)
 (1)  012204  001402                        BEQ     .+6             ;BRANCH IF OK
 (1)  012206  104000                        HLT                     ;STACK POINTER (R4) NOT EQUAL TO #STACK0
 (3)  012210  000420                        420                     ;THE ERROR NUMBER IS 420
 (1)
 (1)  012212  022767  012134  166216        CMP     #RTA55, ANS1    ;CHECK FIS TRAP RETURN ADDRESS
 (1)  012220  001402                        BEQ     .+6             ;BRANCH IF OK
 (1)  012222  104001                        HLT+1                   ;FIS TRAP AT WRONG ADDRESS
 (3)  012224  000421                        421                     ;THE ERROR NUMBER IS 421
 (1)
 (1)  012226  022767  000202  166204        CMP     #202,   ANS2    ;CHECK PS BEFORE FIS TRAP
 (1)  012234  001402                        BEQ     .+6             ;BRANCH IF OK
 (1)  012236  104002                        HLT+2                   ;PS AT FIS TRAP TIME NOT 202
 (3)  012240  000422                        422                     ;THE ERROR NUMBER IS 422
 (1)
 (1)  012242  022767  127652  166172        CMP     #127652,ANS3    ;CHECK DATA FROM THE STACK
 (1)  012250  001402                        BEQ     .+6             ;BRANCH IF OK
 (1)  012252  104004                        HLT+4                   ;DATA ON STACK (127652) CHANGED
 (3)  012254  000423                        423                     ;THE ERROR NUMBER IS 423
 (1)
 (1)  012256  022767  125252  166160        CMP     #125252,ANS4    ;CHECK DATA FROM STACK
 (1)  012264  001402                        BEQ     .+6             ;BRANCH IF OK
 (1)  012266  104004                        HLT+4                   ;DATA ON STACK (125252) CHANGED
 (3)  012270  000424                        424                     ;THE ERROR NUMBER IS 424
 (1)
```

```
(1)  012272  022767  067452  166146          CMP     #067452,ANS5    ;CHECK DATA FROM STACK
(1)  012300  001402                          BEQ     .+6             ;BRANCH IF OK
(1)  012302  104006                          HLT+6                   ;DATA ON STACK (067452) CHANGED
(3)  012304  000425                          425                     ;THE ERROR NUMBER IS 425
(1)
(1)  012306  022767  125252  166134          CMP     #125252,ANS6    ;CHECK DATA FROM STACK
(1)  012314  001402                          BEQ     .+6             ;BRANCH IF OK
(1)  012316  104006                          HLT+6                   ;DATA ON STACK (125252) CHANGED
(3)  012320  000426                          426                     ;THE ERROR NUMBER IS 426
(1)
(1)  012322  122767  000055  166054  END55:  CMPB    #55,    STESTN  ;CHECK THE TEST NUMBER
(1)  012330  001402                          BEQ     .+6             ;BRANCH IF OK
(1)  012332  104000                          HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
(3)  012334  000427                          427                     ;THE ERROR NUMBER IS 427
(1)
(1)
```

```
7231
 (1)                                ;*************************************************************************
 (1)                                ;TEST 56:       FDIV (LSI-11 FLOATING DIVIDE INSTRUCTION)
 (1)                                ;       052525,052525 / 100125,125252 ==> DIVIDE BY ZERO
 (1)                                ;       PS(ON STACK) = 013,     STACK POINTER = R5
 (1)                                ;*************************************************************************
 (1)
 (1)  012336  104400                        SCOPE
 (1)  012340  004567  003020        TST56:  JSR     R5,     PUSHR   ;PUSH 4 WORDS ONTO R5 STACK, SET PRIORITY
 (1)  012344  100125  125252                .WORD   100125,125252   ;SECOND OPERAND ON TOP
 (1)  012350  052525  052525                .WORD   052525,052525   ;FIRST OPERAND ON BOTTOM
 (1)  012354  000047                        .WORD   047             ;PROCESSOR PRIORITY LEVEL
 (1)  012356  012410  000113                .WORD   ISR56, 113      ;FIS TRAP VECTOR
 (1)  012362  012705  000510                MOV     #STACKO,R5      ;SET UP R5 AS STACK POINTER
 (1)
 (1)  012366  000240                        NOP
 (1)  012370  075035                        FDIV    R5              ;FLOATING DIVIDE ON THE R5 STACK
 (1)
 (1)  012372  004767  003020        RTA56:  JSR     PC,     POPR    ;POP THE "ANSWER"
 (1)  012376  010567  166032                MOV     R5,     $SP     ;SAVE STACK POINTER (R5)
 (1)  012402  104002                        HLT+2                   ;FIS TRAP DIDN'T OCCURE!
 (3)  012404  000430                        430                     ;THE ERROR NUMBER IS 430
 (1)  012406  000464                        BR      END56
 (1)
 (1)  012410  004767  003032        ISR56:  JSR     PC,     POPER   ;POP ALL DATA OFF THE STACKS
 (1)  012414  010567  166014                MOV     R5,     $SP     ;SAVE STACK POINTER (R5)
 (1)  012420  122767  000113  166004        CMPB    #113,   $PSW    ;CHECK PS AFTER FIS TRAP
 (1)  012426  001402                        BEQ     .+6             ;BRANCH IF OK
 (1)  012430  104000                        HLT                     ;PS AFTER FIS TRAP NOT EQUAL TO 113
 (3)  012432  000431                        431                     ;THE ERROR NUMBER IS 431
 (1)
 (1)  012434  022767  000510  165772        CMP     #STACKO,$SP     ;CHECK THE STACK POINTER (R5)
 (1)  012442  001402                        BEQ     .+6             ;BRANCH IF OK
 (1)  012444  104000                        HLT                     ;STACK POINTER (R5) NOT EQUAL TO #STACKO
 (3)  012446  000432                        432                     ;THE ERROR NUMBER IS 432
 (1)
 (1)  012450  022767  012372  165760        CMP     #RTA56, ANS1    ;CHECK FIS TRAP RETURN ADDRESS
 (1)  012456  001402                        BEQ     .+6             ;BRANCH IF OK
 (1)  012460  104001                        HLT+1                   ;FIS TRAP AT WRONG ADDRESS
 (3)  012462  000433                        433                     ;THE ERROR NUMBER IS 433
 (1)
 (1)  012464  022767  000013  165746        CMP     #013,   ANS2    ;CHECK PS BEFORE FIS TRAP
 (1)  012472  001402                        BEQ     .+6             ;BRANCH IF OK
 (1)  012474  104002                        HLT+2                   ;PS AT FIS TRAP TIME NOT 013
 (3)  012476  000434                        434                     ;THE ERROR NUMBER IS 434
 (1)
 (1)  012500  022767  100125  165734        CMP     #100125,ANS3    ;CHECK DATA FROM THE STACK
 (1)  012506  001402                        BEQ     .+6             ;BRANCH IF OK
 (1)  012510  104004                        HLT+4                   ;DATA ON STACK (100125) CHANGED
 (3)  012512  000435                        435                     ;THE ERROR NUMBER IS 435
 (1)
 (1)  012514  022767  125252  165722        CMP     #125252,ANS4    ;CHECK DATA FROM STACK
 (1)  012522  001402                        BEQ     .+6             ;BRANCH IF OK
 (1)  012524  104004                        HLT+4                   ;DATA ON STACK (125252) CHANGED
 (3)  012526  000436                        436                     ;THE ERROR NUMBER IS 436
 (1)
```

```
(1)  012530  022767  052525  165710          CMP     #052525,ANS5    ;CHECK DATA FROM STACK
(1)  012536  001402                          BEQ     .+6             ;BRANCH IF OK
(1)  012540  104006                          HLT+6                   ;DATA ON STACK (052525) CHANGED
(3)  012542  000437                          437                     ;THE ERROR NUMBER IS 437
(1)
(1)  012544  022767  052525  165676          CMP     #052525,ANS6    ;CHECK DATA FROM STACK
(1)  012552  001402                          BEQ     .+6             ;BRANCH IF OK
(1)  012554  104006                          HLT+6                   ;DATA ON STACK (052525) CHANGED
(3)  012556  000440                          440                     ;THE ERROR NUMBER IS 440
(1)
(1)  012560  122767  000056  165616  END56:  CMPB    #56,    $TESTN  ;CHECK THE TEST NUMBER
(1)  012566  001402                          BEQ     .+6             ;BRANCH IF OK
(1)  012570  104000                          HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
(3)  012572  000441                          441                     ;THE ERROR NUMBER IS 441
(1)
(1)
```

```
7232
 (1)                                ;*****************************************************************************
 (1)                                ;TEST 57:       FDIV (LSI-11 FLOATING DIVIDE INSTRUCTION)
 (1)                                ;       100052,052525 / 000006,123456 ==> DIVIDE BY ZERO
 (1)                                ;       PS(ON STACK) = 213,     STACK POINTER = SP
 (1)                                ;*****************************************************************************
 (1)
 (1)  012574  104400                            SCOPE
 (1)  012576  004567  002410        TST57:  JSR     R5,     PUSHS   ;PUSH 4 WORDS ONTO STACK, SET PRIORITY
 (1)  012602  000006  123456                .WORD   000006,123456   ;SECOND OPERAND ON TOP
 (1)  012606  100052  052525                .WORD   100052,052525   ;FIRST OPERAND ON BOTTOM
 (1)  012612  000357                        .WORD   357             ;PROCESSOR PRIORITY LEVEL
 (1)  012614  012642  000311                .WORD   ISR57,  311     ;FIS TRAP VECTOR
 (1)
 (1)  012620  000240                        NOP
 (1)  012622  075036                        FDIV    SP              ;FLOATING DIVIDE ON THE STACK
 (1)
 (1)  012624  004767  002422        RTA57:  JSR     PC,     POPS    ;POP THE "ANSWER"
 (1)  012630  104002                        HLT+2                   ;FIS TRAP DIDN'T OCCURE!
 (3)  012632  000442                        442                     ;THE ERROR NUMBER IS 442
 (1)  012634  012706  000600                MOV     #BEGIN, SP      ;RESTORE THE STACK POINTER
 (1)  012640  000464                        BR      END57
 (1)
 (1)  012642  004767  002436        ISR57:  JSR     PC,     POPES   ;POP ALL DATA OFF THE STACK
 (1)  012646  022706  000600                CMP     #BEGIN, SP      ;CHECK THE STACK POINTER
 (1)  012652  001405                        BEQ     ISA57           ;BRANCH IF OK
 (1)  012654  012706  000600                MOV     #BEGIN, SP      ;RESTORE THE STACK POINTER
 (1)  012660  104000                        HLT                     ;STACK POINTER FOULED UP
 (3)  012662  000443                        443                     ;THE ERROR NUMBER IS 443
 (1)  012664  000452                        BR      END57           ;SKIP REST OF TEST
 (1)
 (1)  012666  122767  000311  165536  ISA57:  CMPB    #311,   $PSW    ;CHECK PS AFTER FIS TRAP
 (1)  012674  001402                        BEQ     .+6             ;BRANCH IF OK
 (1)  012676  104000                        HLT                     ;PS AFTER FIS TRAP NOT EQUAL TO 311
 (3)  012700  000444                        444                     ;THE ERROR NUMBER IS 444
 (1)
 (1)  012702  022767  012624  165526        CMP     #RTA57, ANS1    ;CHECK FIS TRAP RETURN ADDRESS
 (1)  012710  001402                        BEQ     .+6             ;BRANCH IF OK
 (1)  012712  104001                        HLT+1                   ;FIS TRAP AT WRONG ADDRESS
 (3)  012714  000445                        445                     ;THE ERROR NUMBER IS 445
 (1)
 (1)  012716  022767  000213  165514        CMP     #213,   ANS2    ;CHECK PS BEFORE FIS TRAP
 (1)  012724  001402                        BEQ     .+6             ;BRANCH IF OK
 (1)  012726  104002                        HLT+2                   ;PS AT FIS TRAP TIME NOT 213
 (3)  012730  000446                        446                     ;THE ERROR NUMBER IS 446
 (1)
 (1)  012732  022767  000006  165502        CMP     #000006,ANS3    ;CHECK DATA FROM THE STACK
 (1)  012740  001402                        BEQ     .+6             ;BRANCH IF OK
 (1)  012742  104004                        HLT+4                   ;DATA ON STACK (000006) CHANGED
 (3)  012744  000447                        447                     ;THE ERROR NUMBER IS 447
 (1)
 (1)  012746  022767  123456  165470        CMP     #123456,ANS4    ;CHECK DATA FROM STACK
 (1)  012754  001402                        BEQ     .+6             ;BRANCH IF OK
 (1)  012756  104004                        HLT+4                   ;DATA ON STACK (123456) CHANGED
 (3)  012760  000450                        450                     ;THE ERROR NUMBER IS 450
 (1)
```

```
(1)  012762  022767  100052  165456          CMP     #100052,ANS5    ;CHECK DATA FROMONG TK
(1)  012770  001402                          BEQ     .+6             ;BRANCH IF OK
(1)  012772  104006                          HLT+6                   ;DATA ON STACK (100052) CHANGED
(3)  012774  000451                          451                     ;THE ERROR NUMBER IS 451
(1)
(1)  012776  022767  052525  165444          CMP     #052525,ANS6    ;CHECK DATA FROM STACK
(1)  013004  001402                          BEQ     .+6             ;BRANCH IF OK
(1)  013006  104006                          HLT+6                   ;DATA ON STACK (052525) CHANGED
(3)  013010  000452                          452                     ;THE ERROR NUMBER IS 452
(1)
(1)  013012  122767  000057  165364  END57:  CMPB    #57,    STESTN  ;CHECK THE TEST NUMBER
(1)  013020  001403                          BEQ     .+6             ;BRANCH IF OK
(1)  013022  104000                          HLT                     ;WREST!  PC MUST HAVE FOULED UP.
(3)  013024  000453                          453                     ;THE ERROR NUMBER IS 453
(1)
(1)
```

```
7237
 (1)
 (1)                    ;*************************************************************************
 (1)                    ;TEST 60:       TEST ALL INSTRUCTION TOGETHER
 (1)                    ;                                032107,065432 * 045670,123456
 (1)                    ;       134343,107070 + ------------------------------  = 137201,115230
 (1)                    ;                               (135252,125252 - 040616,016161)
 (1)                    ;       PS=010, STACK POINTER=R4
 (1)                    ;*************************************************************************
 (1)
 (1)  013026  104400                    SCOPE
 (1)  013030  012704  000532    TST60:  MOV     #STAK10,R4      ;SET STACK POINTER
 (1)  013034  012744  107070            MOV     #107070,-(R4)   ;LOAD DATA ONTO STACK
 (1)  013040  012744  134343            MOV     #134343,-(R4)
 (1)  013044  012744  065432            MOV     #065432,-(R4)
 (1)  013050  012744  032107            MOV     #032107,-(R4)
 (1)  013054  012744  123456            MOV     #123456,-(R4)
 (1)  013060  012744  045670            MOV     #045670,-(R4)
 (1)  013064  012744  125252            MOV     #125252,-(R4)
 (1)  013070  012744  135252            MOV     #135252,-(R4)
 (1)  013074  012744  016161            MOV     #016161,-(R4)
 (1)  013100  012744  040616            MOV     #040616,-(R4)
 (2)  013104                            MTPS    #144            ;SET PROCESSOR STATUS
 (2)  013104  106427                    .WORD   106400!..C
 (1)
 (1)  013110  000240                    NOP
 (1)  013112  075014                    FSUB    R4      ;135252,125252-040616,016161=140616,017434
7238  013114  075034                    FDIV    R4      ;045670,123456/140616,017434=145246,047065
 (1)  013116  075024                    FMUL    R4      ;032107,065432*145246,047065=137201,106137
 (1)  013120  075004                    FADD    R4      ;134343,107070+137201,106137=137201,115230
 (1)
 (2)  013122                            MFPS    $PSW            ;SAVE FINAL PS
 (2)  013122  106767                    .WORD   106700!..C
 (1)  013126  042767  000020  165276    BIC     #20,    $PSW    ;CLR T-BIT
 (1)  013134  012467  165276            MOV     (R4)+,  ANS1    ;SAVE FIRST HALF OF ANSWER
 (1)  013140  012467  165274            MOV     (R4)+,  ANS2    ;SAVE SECOND HALF OF ANSWER
 (1)  013144  010467  165264            MOV     R4,     $SP     ;SAVE STACK POINTER
 (1)  013150  122767  000010  165254    CMPB    #010,   $PSW    ;CHECK PS (EXCEPT T BIT)
 (1)  013156  001402                    BEQ     .+6             ;BRANCH IF OK
 (1)  013160  104000                    HLT                     ;PS NOT EQUAL TO 010
 (3)  013162  000454                    454                     ;THE ERROR NUMBER IS 454
 (1)
 (1)  013164  022767  000532  165242    CMP     #STAK10,$SP     ;CHECK THE STACK POINTER (R4)
 (1)  013172  001402                    BEQ     .+6             ;BRANCH IF OK
 (1)  013174  104000                    HLT                     ;STACK POINTER (R4) NOT EQUAL TO THE
 (3)  013176  000455                    455                     ;THE ERROR NUMBER IS 455
 (1)                                                            ;ADDRESS OF STAK10
 (1)
 (1)  013200  022767  137201  165230    CMP     #137201,ANS1    ;CHECK FIRST HALF OF ANSWER
 (1)  013206  001402                    BEQ     .+6             ;BRANCH IF OK
 (1)  013210  104002                    HLT+2                   ;ANS1 NOT EQUAL TO 137201
 (3)  013212  000456                    456                     ;THE ERROR NUMBER IS 456
 (1)
 (1)  013214  022767  115230  165216    CMP     #115230,ANS2    ;CHECK SECOND HALF OF ANSWER
 (1)  013222  001402                    BEQ     .+6             ;BRANCH IF OK
 (1)  013224  104002                    HLT+2                   ;ANS2 NOT EQUAL TO 115230
 (3)  013226  000457                    457                     ;THE ERROR NUMBER IS 457
```

```
    (1)
    (1)  013230  122767  000060  165146  END60:  CMPB    #60,    $TESTN  ;CHECK THE TEST NUMBER
    (1)  013236  001402                          BEQ     .+6             ;BRANCH IF OK
    (1)  013240  104000                          HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
    (3)  013242  000460                          460                     ;THE ERROR NUMBER IS 460
    (1)
```

```
7243
 (1)                                ;*****************************************************************************
 (1)                                ;TEST 61:       TEST THAT STACK POINTER ADDRESS ERROR CAUSES ABORT
 (1)                                ;       INSTRUCTION = FADD,     STACK POINTER = R2
 (1)                                ;*****************************************************************************
 (1)
 (1)  013244  104400                            SCOPE
 (1)  013246  012737  013336  000004  TST61:    MOV     #ISR61, @#4     ;SET UP ADDRESS TRAP VECTOR
 (1)  013254  012737  000340  000006            MOV     #340,   @#6
 (1)  013262  004567  002076                    JSR     R5,     PUSHR   ;PUSH 4 WORDS ONTO R2 STACK, SET PRIORITY
 (1)  013266  070707  016161                    .WORD   070707,016161   ;SECOND OPERAND ON TOP
 (1)  013272  146314  143434                    .WORD   146314,143434   ;FIRST OPERAND ON BOTTOM
 (1)  013276  000143                            .WORD   143             ;PROCESSOR PRIORITY LEVEL
 (1)  013300  015634  000340                    .WORD   TRAPER, 340     ;FIS TRAP VECTOR
 (2)  013304                                    MTPS    #143            ;SET PROCESSOR STATUS
 (2)  013304  106427                            .WORD   106400!..C
 (1)  013310  012702  177777                    MOV     #177777,R2      ;SET UP R2 AS STACK POINTER
 (1)
 (1)  013314  000240                            NOP
 (1)  013316  075002                            FADD    R2              ;FLOATING ADD ON THE R2 STACK
 (1)
 (1)  013320  004767  002072            RTA61:  JSR     PC,     POPR    ;POP THE "ANSWER"
 (1)  013324  010267  165104                    MOV     R2,     $SP     ;SAVE STACK POINTER (R2)
 (1)  013330  104002                            HLT+2                   ;FIS TRAP DIDN'T OCCURE!
 (3)  013332  000461                            461                     ;THE ERROR NUMBER IS 461
 (1)  013334  000464                            BR      END61
 (1)
 (1)  013336  004767  002104            ISR61:  JSR     PC,     POPER   ;POP ALL DATA OFF THE STACKS
 (1)  013342  010267  165066                    MOV     R2,     $SP     ;SAVE STACK POINTER (R2)
 (1)  013346  122767  000340  165056            CMPB    #340,   SPSW    ;CHECK PS AFTER ADR. ERR. TRAP
 (1)  013354  001402                            BEQ     .+6             ;BRANCH IF OK
 (1)  013356  104000                            HLT                     ;PS AFTER TRAP NOT EQUAL TO 340
 (3)  013360  000462                            462                     ;THE ERROR NUMBER IS 462
 (1)
 (1)  013362  022767  177777  165044            CMP     #177777,$SP     ;CHECK THE STACK POINTER (R2)
 (1)  013370  001402                            BEQ     .+6             ;BRANCH IF OK
 (1)  013372  104000                            HLT                     ;STACK POINTER (R2) NOT EQUAL TO #177777
 (3)  013374  000463                            463                     ;THE ERROR NUMBER IS 463
 (1)
 (1)  013376  022767  013320  165032            CMP     #RTA61, ANS1    ;CHECK FIS TRAP RETURN ADDRESS
 (1)  013404  001402                            BEQ     .+6             ;BRANCH IF OK
 (1)  013406  104001                            HLT+1                   ;FIS TRAP AT WRONG ADDRESS
 (3)  013410  000464                            464                     ;THE ERROR NUMBER IS 464
 (1)
 (1)  013412  022767  000151  165020            CMP     #151,   ANS2    ;CHECK PS BEFORE FIS TRAP
 (1)  013420  001402                            BEQ     .+6             ;BRANCH IF OK
 (1)  013422  104002                            HLT+2                   ;PS AT FIS TRAP TIME NOT 151
 (3)  013424  000465                            465                     ;THE ERROR NUMBER IS 465
 (1)
 (1)  013426  022767  070707  165006            CMP     #070707,ANS3    ;CHECK DATA FROM THE STACK
 (1)  013434  001402                            BEQ     .+6             ;BRANCH IF OK
 (1)  013436  104004                            HLT+4                   ;DATA ON STACK (070707) CHANGED
 (3)  013440  000466                            466                     ;THE ERROR NUMBER IS 466
 (1)
 (1)  013442  022767  016161  164774            CMP     #016161,ANS4    ;CHECK DATA FROM STACK
 (1)  013450  001402                            BEQ     .+6             ;BRANCH IF OK
```

F06


```
(1)  013452  104004                          HLT+4                ;DATA ON STACK (016161) CHANGED
(3)  013454  000467                          467                  ;THE ERROR NUMBER IS 467
(1)
(1)  013456  022767  146314  164762          CMP     #146314,ANS5 ;CHECK DATA FROM STACK
(1)  013464  001402                          BEQ     .+6          ;BRANCH IF OK
(1)  013466  104006                          HLT+6                ;DATA ON STACK (146314) CHANGED
(3)  013470  000470                          470                  ;THE ERROR NUMBER IS 470
(1)
(1)  013472  022767  143434  164750          CMP     #143434,ANS6 ;CHECK DATA FROM STACK
(1)  013500  001402                          BEQ     .+6          ;BRANCH IF OK
(1)  013502  104006                          HLT+6                ;DATA ON STACK (143434) CHANGED
(3)  013504  000471                          471                  ;THE ERROR NUMBER IS 471
(1)  013506  122767  000061  164670  END61:  CMPB    #61,    $TESTN ;CHECK THE TEST NUMBER
(1)  013514  001402                          BEQ     .+6          ;BRANCH IF OK
(1)  013516  104000                          HLT                  ;WRONG TEST!  PC MUST HAVE FOULED UP.
(3)  013520  000472                          472                  ;THE ERROR NUMBER IS 472
```

```
7244
 (1)                        ;*****************************************************************************
 (1)                        ;TEST 62:       TEST THAT STACK POINTER ADDRESS ERROR CAUSES ABORT
 (1)                        ;       INSTRUCTION = FMUL,     STACK POINTER = R5
 (1)                        ;*****************************************************************************
 (1)
 (1)  013522  104400                    SCOPE
 (1)  013524  012737  013614  000004 TST62: MOV     #ISR62, @#4     ;SET UP ADDRESS TRAP VECTOR
 (1)  013532  012737  000340  000006       MOV     #340,   @#6
 (1)  013540  004567  001620               JSR     R5,     PUSHR   ;PUSH 4 WORDS ONTO R5 STACK, SET PRIORITY
 (1)  013544  065432  123456               .WORD   065432,123456   ;SECOND OPERAND ON TOP
 (1)  013550  037654  032107               .WORD   037654,032107   ;FIRST OPERAND ON BOTTOM
 (1)  013554  000202                       .WORD   202             ;PROCESSOR PRIORITY LEVEL
 (1)  013556  015634  000340               .WORD   TRAPER, 340     ;FIS TRAP VECTOR
 (2)  013562                               MTPS    #202            ;SET PROCESSOR STATUS
 (2)  013562  106427                       .WORD   106400!..C
 (1)  013566  012705  160000               MOV     #160000,R5      ;SET UP R5 AS STACK POINTER
 (1)
 (1)  013572  000240                       NOP
 (1)  013574  075025                       FMUL    R5              ;FLOATING MULTIPLY ON THE R5 STACK
 (1)
 (2)  013576                        RTA62: MFPS    $PSW            ;SAVE THE PSW
 (2)  013576  106767                       .WORD   106700!..C
 (1)  013602  010567  164626               MOV     R5,     $SP     ;SAVE STACK POINTER (R5)
 (1)  013606  104000                       HLT                     ;FIS TRAP DIDN'T OCCURE!
 (3)  013610  000473                       473                     ;THE ERROR NUMBER IS 473
 (1)  013612  000434                       BR      END62
 (1)
 (1)  013614  004767  001626        ISR62: JSR     PC,     POPER   ;POP ALL DATA OFF THE STACKS
 (1)  013620  010567  164610               MOV     R5,     $SP     ;SAVE STACK POINTER (R5)
 (1)  013624  122767  000340  164600       CMPB    #340,   $PSW    ;CHECK PS AFTER ADR. ERR. TRAP
 (1)  013632  001402                       BEQ     .+6             ;BRANCH IF OK
 (1)  013634  104000                       HLT                     ;PS AFTER TRAP NOT EQUAL TO 340
 (3)  013636  000474                       474                     ;THE ERROR NUMBER IS 474
 (1)
 (1)  013640  022767  160000  164566       CMP     #160000,$SP     ;CHECK THE STACK POINTER (R5)
 (1)  013646  001402                       BEQ     .+6             ;BRANCH IF OK
 (1)  013650  104000                       HLT                     ;STACK POINTER (R5) NOT EQUAL TO #160000
 (3)  013652  000475                       475                     ;THE ERROR NUMBER IS 475
 (1)
 (1)  013654  022767  013576  164554       CMP     #RTA62, ANS1    ;CHECK FIS TRAP RETURN ADDRESS
 (1)  013662  001402                       BEQ     .+6             ;BRANCH IF OK
 (1)  013664  104001                       HLT+1                   ;FIS TRAP AT WRONG ADDRESS
 (3)  013666  000476                       476                     ;THE ERROR NUMBER IS 476
 (1)
 (1)  013670  022767  000210  164542       CMP     #210,   ANS2    ;CHECK PS BEFORE FIS TRAP
 (1)  013676  001402                       BEQ     .+6             ;BRANCH IF OK
 (1)  013700  104002                       HLT+2                   ;PS AT FIS TRAP TIME NOT 210
 (3)  013702  000477                       477                     ;THE ERROR NUMBER IS 477
 (1)
 (1)  013704  122767  000062  164472 END62: CMPB    #62,    $TESTN  ;CHECK THE TEST NUMBER
 (1)  013712  001402                       BEQ     .+6             ;BRANCH IF OK
 (1)  013714  104000                       HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
 (3)  013716  000500                       500                     ;THE ERROR NUMBER IS 500
```

```
 7245  013720  012737  000006  000004          MOV     #6,     @#4     ;RESTORE TIME-OUT VECTOR
 7246  013726  005037  000006                  CLR     @#6
 7250  013732  012767  000003  164576          MOV     #3,     TIMES   ;REDUCE NUMBER OF ITTERATIONS
 7251
  (1)                                ;*************************************************************************
  (1)                                ;TEST 63:       TEST THAT FIS ABORTS PROPERLY WHEN INTERUPTED
  (1)                                ;       035700,143235 + 000177,134543 = 035700,143235
  (1)                                ;       PS = .PS,       STACK POINTER = R1
  (1)                                ;*************************************************************************
  (1)
  (1)   013740  104400                          SCOPE
  (1)   013742  132737  000040  000421  TST63:  BITB    #40,@#$ENVM
  (1)   013750  001170                          BNE     END63+2         ;EXIT THIS TEST IF BIT 5 OF $ENVM IS HIGH
  (1)   013752  013704  000546                  MOV     @#TTYOUT,R4
  (1)   013756  012724  014050                  MOV     #ISR63, (R4)+   ;SET UP TELEPRINTER INTERUPT VECTOR
  (1)   013762  012714  000340                  MOV     #340,   (R4)
  (1)   013766  000004  000473                  TYPE,   RETURN+1        ;RETURN+1 CAN BE REPLACED WITH THE ADDRESS OF RETURN
  (1)                                                                   ;TO TYPE CARRIAGE RETURN, LINE FEED
  (1)   013772  012767  014000  164470          MOV     .+6,    LAD$    ;RESET LOOP ADDRESS
  (1)   014000  004567  001360                  JSR     R5,     PUSHR   ;PUSH 4 WORDS ONTO R1 STACK, SET PRIORITY
  (1)   014004  000177  134543                  .WORD   000177,134543   ;SECOND OPERAND ON TOP
  (1)   014010  035700  143235                  .WORD   035700,143235   ;FIRST OPERAND ON BOTTOM
  (1)   014014  000143                          .WORD   143             ;PROCESSOR PRIORITY LEVEL
  (1)   014016  015634  000340                  .WORD   TRAPER, 340     ;FIS TRAP VECTOR
  (1)   014022  012701  000510                  MOV     #STACK0,R1      ;SET UP STACK POINTER
  (1)   014026  012767  000030  164500          MOV     #30,TEMP
  (1)   014034  112777  000100  164506          MOVB    #100,   @$TPS   ;SET TTY INTERUPT ENABLE
  (1)
  (1)   014042  075001                  RTA63:  FADD    R1              ;FLOATING ADD ON THE STACK
  (1)   014044  024141                          CMP     -(R1),  -(R1)   ;RESET THE STACK POINTER FOR NEXT PASS
  (1)   014046  000775                          BR      RTA63           ;REPEAT UNTIL INTERUPTED
  (1)
  (1)   014050  105077  164474          ISR63:  CLRB    @$TPS           ;CLEAR THE INTERUPT ENABLE
  (1)   014054  022716  014042                  CMP     #RTA63, (SP)    ;CHECK IF INTERUPT AT FIS INSTR.
  (1)   014060  001421                          BEQ     3$              ;BRANCH IF IT DID
  (1)   014062  022766  014042  000004          CMP     #RTA63, 4(SP)   ;CHECK FOR INTERUPT WITH T-BIT SET
  (1)   014070  001420                          BEQ     4$              ;BRANCH IF IT DID
  (1)   014072  112777  000015  164452  1$:     MOVB    #15,    @$TPB   ;CONTINUE TO TYPE "CR"
  (1)   014100  105777  164444          2$:     TSTB    @$TPS           ;LOOP HERE UNTILL DONE BIT COMES ON
  (1)   014104  100375                          BPL     2$
  (1)   014106  112777  000015  164436          MOVB    #15,    @$TPB   ;TYPE ANOTHER "CR"
  (1)   014114  012777  000100  164426          MOV     #100,   @$TPS   ;SET TTY INTERUPT ENABLE
  (1)   014122  000002                          RTI
  (1)
  (1)   014124  004767  001316          3$:     JSR     PC,     POPER   ;SAVE ALL THE STUFF ON THE STACK
  (1)   014130  000403                          BR      5$
  (1)
  (1)   014132  022626                  4$:     CMP     (SP)+,  (SP)+   ;RESET THE STACK TO IGNORE THE TRACE TRAP
  (1)   014134  004767  001312                  JSR     PC,     POPER1  ;POP ALL THE STUFF OFF THE STACK
  (1)   014140  005746                  5$:     TST     -(SP)           ;SAVE PSW FOR FUTURE RTI
  (1)   014142  012746  014042                  MOV     #RTA63,-(SP)    ;PLACE THE RTI ADDRESS BACK IN SP
  (1)   014146  022706  000574                  CMP     #BEGIN-4,SP     ;CHECK THE STACK POINTER
  (1)   014152  001407                          BEQ     6$              ;BRANCH IF OK
  (1)   014154  010667  164254                  MOV     SP,     $SP     ;SAVE FOR TYPING
  (1)   014160  012706  000574                  MOV     #BEGIN-4,SP     ;RESTORE THE STACK POINTER
  (1)   014164  104000                          HLT                     ;STACK POINTER FOULED UP
```

```
(3)  014166  000501                          501                ;THE ERROR NUMBER IS 501
(1)  014170  000457                          BR      END63      ;SKIP REST OF TEST
(1)
(1)  014172  010167  164236          6$:     MOV     R1,    $SP ;SAVE STACK POINTER
(1)  014176  122767  000344  164226          CMPB    #344,  $PSW;CHECK PS AFTER INTERUPT
(1)  014204  001402                          BEQ     .+6        ;BRANCH IF OK
(1)  014206  104000                          HLT                ;PS AFTER INTERUPT NOT EQUAL TO LVLA
(3)  014210  000502                          502                ;THE ERROR NUMBER IS 502
(1)
(1)  014212  022767  000510  164214          CMP     #STACK0,$SP;CHECK THE STACK POINTER (R1)
(1)  014220  001402                          BEQ     .+6        ;BRANCH IF OK
(1)  014222  104000                          HLT                ;STACK POINTER (R1) NOT EQUAL TO #STACK0
(3)  014224  000503                          503                ;THE ERROR NUMBER IS 503
(1)
(1)  014226  022767  014042  164202          CMP     #RTA63, ANS1;CHECK FIS TRAP RETURN ADDRESS
(1)  014234  001402                          BEQ     .+6        ;BRANCH IF OK
(1)  014236  104001                          HLT+1              ;FIS TRAP AT WRONG ADDRESS
(3)  014240  000504                          504                ;THE ERROR NUMBER IS 504
(1)
(1)
(1)  014242  022767  000177  164172          CMP     #000177,ANS3;CHECK DATA FROM THE STACK
(1)  014250  001402                          BEQ     .+6        ;BRANCH IF OK
(1)  014252  104004                          HLT+4              ;DATA ON STACK (000177) CHANGED
(3)  014254  000505                          505                ;THE ERROR NUMBER IS 505
(1)
(1)  014256  022767  134543  164160          CMP     #134543,ANS4;CHECK DATA FROM STACK
(1)  014264  001402                          BEQ     .+6        ;BRANCH IF OK
(1)  014266  104004                          HLT+4              ;DATA ON STACK (134543) CHANGED
(3)  014270  000506                          506                ;THE ERROR NUMBER IS 506
(1)
(1)  014272  022767  035700  164146          CMP     #035700,ANS5;CHECK DATA FROM STACK
(1)  014300  001402                          BEQ     .+6        ;BRANCH IF OK
(1)  014302  104006                          HLT+6              ;DATA ON STACK (035700) CHANGED
(3)  014304  000507                          507                ;THE ERROR NUMBER IS 507
(1)
(1)  014306  022767  143235  164134          CMP     #143235,ANS6;CHECK DATA FROM STACK
(1)  014314  001402                          BEQ     .+6        ;BRANCH IF OK
(1)  014316  104006                          HLT+6              ;DATA ON STACK (143235) CHANGED
(3)  014320  000510                          510                ;THE ERROR NUMBER IS 510
(1)
(1)  014322  005367  164206                  DEC     TEMP       ;STAY IN THE LOOP FOR 30 TIMES
(1)  014326  001261                          BNE     1$
(1)
(1)  014330  022626                  END63:  CMP     (SP)+, (SP)+;RESTORE STACK POINTER TO 500
(1)  014332  122767  000063  164044          CMPB    #63,   $TESTN;CHECK THE TEST NUMBER
(1)  014340  001402                          BEQ     .+6        ;BRANCH IF OK
(1)  014342  104000                          HLT                ;WRONG TEST!  PC MUST HAVE FOULED UP.
(3)  014344  000511                          511                ;THE ERROR NUMBER IS 511
(2)  014346                                  MTPS    #340
(2)  014346  106427                          .WORD   106400!..C
(1)
```

```
7252
 (1)                        ;*************************************************************************
 (1)                        ;TEST 64:       TEST THAT FIS ABORTS PROPERLY WHEN INTERUPTED
 (1)                        ;       107070,070707 * 040200,000000 = 107070,070707
 (1)                        ;       PS = .PS,       STACK POINTER = R0
 (1)                        ;*************************************************************************
 (1)
 (1) 014352 104400                  SCOPE
 (1) 014354 132737 000040 000421 TST64: BITB    #40,@#$ENVM
 (1) 014362 001167                  BNE     END64+2         ;EXIT THIS TEST IF BIT 5 OF $ENVM IS HIGH
 (1) 014364 013704 000546           MOV     @#TTYOUT,R4
 (1) 014370 012724 014462           MOV     #ISR64, (R4)+   ;SET UP TELEPRINTER INTERUPT VECTOR
 (1) 014374 012714 000340           MOV     #340,   (R4)
 (1) 014400 000004 000473           TYPE,   RETURN+1        ;RETURN+1 CAN BE REPLACED WITH THE ADDRESS OF RETURN
 (1)                                                        ;TO TYPE CARRIAGE RETURN, LINE FEED
 (1) 014404 012767 014412 164056    MOV     #.+6,   LAD$    ;RESET LOOP ADDRESS
 (1) 014412 004567 000746           JSR     R5,     PUSHR   ;PUSH 4 WORDS ONTO R0 STACK, SET PRIORITY
 (1) 014416 040200 000000           .WORD   040200,000000   ;SECOND OPERAND ON TOP
 (1) 014422 107070 070707           .WORD   107070,070707   ;FIRST OPERAND ON BOTTOM
 (1) 014426 000100                  .WORD   100             ;PROCESSOR PRIORITY LEVEL
 (1) 014430 015634 000340           .WORD   TRAPER, 340     ;FIS TRAP VECTOR
 (1) 014434 012700 000510           MOV     #STACK0,R0      ;SET UP STACK POINTER
 (1) 014440 012767 000030 164066    MOV     #30,TEMP
 (1) 014446 112777 000100 164074    MOVB    #100,   @$TPS   ;SET TTY INTERUPT ENABLE
 (1)
 (1) 014454 075020          RTA64:  FMUL    R0              ;FLOATING MULTIPLY ON THE STACK
 (1) 014456 024040                  CMP     -(R0),  -(R0)   ;RESET THE STACK POINTER FOR NEXT PASS
 (1) 014460 000775                  BR      RTA64           ;REPEAT UNTIL INTERUPTED
 (1)
 (1) 014462 105077 164062   ISR64:  CLRB    @$TPS           ;CLEAR THE INTERUPT ENABLE
 (1) 014466 022716 014454           CMP     #RTA64, (SP)    ;CHECK IF INTERUPT AT FIS INSTR.
 (1) 014472 001421                  BEQ     3$              ;BRANCH IF IT DID
 (1) 014474 022766 014454 000004    CMP     #RTA64, 4(SP)   ;CHECK FOR INTERUPT WITH T-BIT SET
 (1) 014502 001420                  BEQ     4$              ;BRANCH IF IT DID
 (1) 014504 112777 000015 164040 1$: MOVB   #15,    @$TPB   ;CONTINUE TO TYPE "CR"
 (1) 014512 105777 164032        2$: TSTB   @$TPS           ;LOOP HERE UNTILL DONE BIT COMES ON
 (1) 014516 100375                  BPL     2$
 (1) 014520 112777 000015 164024    MOVB    #15,    @$TPB   ;TYPE ANOTHER "CR"
 (1) 014526 012777 000100 164014    MOV     #100,   @$TPS   ;SET TTY INTERUPT ENABLE
 (1) 014534 000002                  RTI
 (1)
 (1) 014536 004767 000704        3$: JSR     PC,     POPER   ;SAVE ALL THE STUFF ON THE STACK
 (1) 014542 000403                  BR      5$
 (1)
 (1) 014544 022626               4$: CMP     (SP)+,  (SP)+   ;RESET THE STACK TO IGNORE THE TRACE TRAP
 (1) 014546 004767 000700            JSR     PC,     POPER1  ;POP ALL THE STUFF OFF THE STACK
 (1) 014552 005746               5$: TST     -(SP)           ;SAVE PSW FOR FUTURE RTI
 (1) 014554 012746 014454            MOV     #RTA64,-(SP)    ;PLACE THE RTI ADDRESS BACK IN SP
 (1) 014560 022706 000574            CMP     #BEGIN-4,SP     ;CHECK THE STACK POINTER
 (1) 014564 001407                   BEQ     6$              ;BRANCH IF OK
 (1) 014566 010667 163642            MOV     SP,     $SP     ;SAVE FOR TYPING
 (1) 014572 012706 000574            MOV     #BEGIN-4,SP     ;RESTORE THE STACK POINTER
 (1) 014576 104000                   HLT                     ;STACK POINTER FOULED UP
 (3) 014600 000512                   512                     ;THE ERROR NUMBER IS 512
 (1) 014602 000456                   BR      END64           ;SKIP REST OF TEST
 (1)
```

```
(1)  014604  010067  163624          6$:   MOV     R0,     $SP     ;SAVE STACK POINTER
(1)  014610  122767  000344  163614        CMPB    #344,   $PSW    ;CHECK PS AFTER INTERUPT
(1)  014616  001402                        BEQ     .+6             ;BRANCH IF OK
(1)  014620  104000                        HLT                     ;PS AFTER INTERUPT NOT EQUAL TO LVLA
(3)  014622  000513                        513                     ;THE ERROR NUMBER IS 513
(1)
(1)  014624  022767  000510  163602        CMP     #STACK0,$SP     ;CHECK THE STACK POINTER (R0)
(1)  014632  001402                        BEQ     .+6             ;BRANCH IF OK
(1)  014634  104000                        HLT                     ;STACK POINTER (R0) NOT EQUAL TO #STACK0
(3)  014636  000514                        514                     ;THE ERROR NUMBER IS 514
(1)
(1)  014640  022767  014454  163570        CMP     #RTA64, ANS1    ;CHECK FIS TRAP RETURN ADDRESS
(1)  014646  001402                        BEQ     .+6             ;BRANCH IF OK
(1)  014650  104001                        HLT+1                   ;FIS TRAP AT WRONG ADDRESS
(3)  014652  000515                        515                     ;THE ERROR NUMBER IS 515
(1)
(1)
(1)  014654  022767  040200  163560        CMP     #040200,ANS3    ;CHECK DATA FROM THE STACK
(1)  014662  001402                        BEQ     .+6             ;BRANCH IF OK
(1)  014664  104004                        HLT+4                   ;DATA ON STACK (040200) CHANGED
(3)  014666  000516                        516                     ;THE ERROR NUMBER IS 516
(1)
(1)  014670  005767  163550                TST     ANS4            ;CHECK DATA FROM STACK
(1)  014674  001402                        BEQ     .+6             ;BRANCH IF OK
(1)  014676  104004                        HLT+4                   ;DATA ON STACK (000000) CHANGED
(3)  014700  000517                        517                     ;THE ERROR NUMBER IS 517
(1)
(1)  014702  022767  107070  163536        CMP     #107070,ANS5    ;CHECK DATA FROM STACK
(1)  014710  001402                        BEQ     .+6             ;BRANCH IF OK
(1)  014712  104006                        HLT+6                   ;DATA ON STACK (107070) CHANGED
(3)  014714  000520                        520                     ;THE ERROR NUMBER IS 520
(1)
(1)  014716  022767  070707  163524        CMP     #070707,ANS6    ;CHECK DATA FROM STACK
(1)  014724  001402                        BEQ     .+6             ;BRANCH IF OK
(1)  014726  104006                        HLT+6                   ;DATA ON STACK (070707) CHANGED
(3)  014730  000521                        521                     ;THE ERROR NUMBER IS 521
(1)
(1)  014732  005367  163576                DEC     TEMP            ;STAY IN THE LOOP FOR 30 TIMES
(1)  014736  001262                        BNE     1$
(1)
(1)  014740  022626                END64:  CMP     (SP)+,  (SP)+   ;RESTORE STACK POINTER TO 500
(1)  014742  122767  000064  163434        CMPB    #64,    $TESTN  ;CHECK THE TEST NUMBER
(1)  014750  001402                        BEQ     .+6             ;BRANCH IF OK
(1)  014752  104000                        HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
(3)  014754  000522                        522                     ;THE ERROR NUMBER IS 522
(2)  014756                                MTPS    #340
(2)  014756  106427                        .WORD   106400!..C
(1)
```

```
 7253  014762  012767  000377  163546          MOV     #377,   TIMES   ;SET NUMBER OF ITERATIONS TO 377
 7254  014770  010477  163552                  MOV     R4,     @TTYOUT ;RESTORE TTY INTERUPT VECTOR
 7255  014774  005014                          CLR     (R4)
 7256
```

```
7258                                    ;*****************************************************************
 (1)
 (1)                                    .SBTTL  END OF PASS ROUTINE
 (1)
 (1)                                    ;*INCREMENT THE PASS NUMBER ($PASS)
 (1)                                    ;*TYPE "END PASS"
 (1)                                    ;*IF THERES A MONITOR GO TO IT
 (1)                                    ;*IF THERE ISN'T JUMP TO BEGIN
 (1)                                    ;*IF IT IS DESIRED TO HAVE A BELL INDICATE THE "END OF PASS" LOCATION
 (1)                                    ;*SENDMG CAN BE CHANGED TO 7.
 (1)
 (1)  014776                            $EOP:
 (1)  014776  104400                            SCOPE
 (1)  015000  005267  163402                    INC     $PASS           ;;INCREMENT THE PASS NUMBER
 (1)  015004  042767  100000  163374            BIC     #100000,$PASS   ;;DON'T ALLOW A NEG. NUMBER
 (1)  015012  005327                            DEC     (PC)+           ;;LOOP?
 (1)  015014  000001                    $EOPCT: .WORD   1
 (1)  015016  003015                            BGT     $DOAGN          ;;YES
 (1)  015020  012737                            MOV     (PC)+,@(PC)+    ;;RESTORE COUNTER
 (1)  015022  000001                    $ENDCT: .WORD   1
 (1)  015024  015014                            $EOPCT
 (1)  015026  000004  015056                    TYPE    ,$ENDMG         ;;TYPE "END PASS"
 (1)  015032                            $GET42:
 (1)
 (1)  015032  013700  000042                    MOV     @#42,R0         ;;GET MONITOR ADDRESS
 (1)  015036  001405                            BEQ     $DOAGN          ;;BRANCH IF NO MONITOR
 (1)  015040  000005                            RESET                   ;;CLEAR THE WORLD
 (1)  015042  004710                    $ENDAD: JSR     PC,(R0)         ;;GO TO MONITOR
 (1)  015044  000240                            NOP                     ;;SAVE ROOM
 (1)  015046  000240                            NOP                     ;;FOR
 (1)  015050  000240                            NOP                     ;;ACT11
 (1)  015052                            $DOAGN:
 (1)  015052  000137  000600                    JMP     @#BEGIN         ;;RETURN
 (1)  015056  005015  047105  020104    $ENDMG: .ASCII  <15><12>/END PASS/
 (1)  015064  040520  051523
 (1)  015070     377     377     000    $ENULL: .BYTE   -1,-1,0         ;;NULL CHARACTER STRING
 (1)          015074                            .EVEN
7259
7264  015022  000001                    ENDCT:  1
```

```
 7268                                   ;*****************************************************************
  (1)
  (1)                                   .SBTTL  SCOPE ROUTINE
  (1)
  (1)  015074  032737  000400  000422  SCOPE$:  BIT     #SW08,@#$SWREG  ;KILL LDUB OR LOOP ON SPEC. TEST
  (1)  015102  001404                           BEQ     1$
  (1)  015104  123767  000422  163272           CMPB    @#$SWREG,$TESTN ;ON RIGHT TEST?   *SW7-0*
  (1)  015112  001431                           BEQ     OVER$
  (1)  015114  032737  040000  000422  1$:     BIT     #SW14,@#$SWREG  ;LOOP ON TEST
  (1)  015122  001023                           BNE     KIT$
  (1)  015124  032737  004000  000422           BIT     #SW11,@#$SWREG  ;KILL ITERATIONS
  (1)  015132  001412                           BEQ     SVLAD$
  (1)  015134  105767  163343                   TSTB    $ICNT
  (1)  015140  001404                           BEQ     2$              ;BRANCH IF FIRST
  (1)  015142  126767  163370  163333           CMPB    TIMES,$ICNT     ;DONE?
  (1)  015150  001010                           BNE     KIT$            ;BRANCH IF NOT
  (1)  015152  112767  000001  163323  2$:     MOVB    #1,$ICNT        ;FIRST ITERATION
  (1)  015160  105267  163220          SVLAD$: INCB    $TESTN          ;COUNT TEST NUMBERS
  (1)  015164  011667  163300                   MOV     (6),LAD$        ;SAVE LOOP ADDRESS
  (1)  015170  000002                           RTI                     ;RETURN
  (1)
  (1)  015172  105267  163305          KIT$:   INCB    $ICNT
  (1)  015176  005767  163266          OVER$:  TST     LAD$            ;FIRST ONE?
  (1)  015202  001766                           BEQ     SVLAD$
  (1)  015204  016716  163260                   MOV     LAD$,(6)        ;FUDGE RETURN ADDRESS
  (1)  015210  000002                           RTI                     ;FIXES PS
  (1)
```

```
7273
7274                                    ;SUBROUTINE TO PUSH 4 WORDS ONTO THE STACK
7275
7276  015212  005726                    PUSHS:  TST     (SP)+           ;POP STACK BY 1
7277  015214  062705  000010                    ADD     #10,    R5      ;POINT TO END OF DATA
7278  015220  014546                            MOV     -(R5),  -(SP)   ;PUSH DATA ONTO THE STACK
7279  015222  014546                            MOV     -(R5),  -(SP)   ;PUSH DATA ONTO THE STACK
7280  015224  014546                            MOV     -(R5),  -(SP)   ;PUSH DATA ONTO THE STACK
7281  015226  014546                            MOV     -(R5),  -(SP)   ;PUSH DATA ONTO THE STACK
7282  015230  062705  000010                    ADD     #10,    R5      ;POINT TO END OF DATA
7286  015234                                    MTPS    (R5)+           ;SET THE PROCESSOR STATUS
 (1)  015234  106425                            .WORD   106400!..C
7290  015236  005205                            INC     R5
7291  015240  012577  163220                    MOV     (R5)+,  @FISVEC ;SET UP FIS ERROR TRAP VECTOR
7292  015244  012577  163216                    MOV     (R5)+,  @FISLVL ;TRAP STATUS
7293  015250  000115                            JMP     (R5)            ;RETURN
7294
7295
7296                                    ;SUBROUTINE TO POP 2 WORDS OFF THE STACK
7297                                    ;ALSO SAVES THE PROCESSOR STATUS WORD (EXCEPT T BIT)
7298
7302  015252                            POPS:   MFPS    $PSW            ;SAVE PROCESSOR STATUS WORD
 (1)  015252  106767                            .WORD   106700!..C
7306  015256  042767  000020  163146            BIC     #20,    $PSW    ;CLEAR T-BIT
7307  015264  012604                            MOV     (SP)+,  R4      ;SAVE RTS ADDRESS
7308  015266  012667  163144                    MOV     (SP)+,  ANS1    ;SAVE THE ANSWER
7309  015272  012667  163142                    MOV     (SP)+,  ANS2    ;
7310  015276  010667  163132                    MOV     SP,     $SP     ;SAVE THE STACK POINTER
7311  015302  000114                            JMP     (R4)            ;RETURN
7312
7313
7314                                    ;SUBROUTNE TO POP 6 WORDS OFF THE STACK.
7315                                    ;THE FIRST TWO WERE PUT ON BY THE ERROR TRAP,
7316                                    ;THE LAST FOUR WERE THE ORIGINAL INPUT DATA.
7317                                    ;ALSO SAVES THE PS AND STACK POINTER.
7318
7322  015304                            POPES:  MFPS    $PSW            ;SAVE PROCESSOR STATUS WORD
 (1)  015304  106767                            .WORD   106700!..C
7326  015310  012604                            MOV     (SP)+,  R4      ;SAVE RTS ADDRESS
7327  015312  012667  163120                    MOV     (SP)+,  ANS1    ;SAVE RTI ADDRESS
7328  015316  011667  163116                    MOV     (SP),   ANS2    ;SAVE RTI STATUS
7329  015322  042767  000020  163110            BIC     #20,    ANS2    ;CLEAR THE T-BIT
7330  015330  012746  015336                    MOV     #1$,    -(SP)
7331  015334  000002                            RTI                     ;RESTORE THE PROCESSOR STATUS
7332  015336  012667  163100            1$:     MOV     (SP)+,  ANS3    ;SAVE DATA
7333  015342  012667  163076                    MOV     (SP)+,  ANS4    ;
7334  015346  012667  163074                    MOV     (SP)+,  ANS5    ;
7335  015352  012667  163072                    MOV     (SP)+,  ANS6    ;
7336  015356  010667  163052                    MOV     SP,     $SP     ;SAVE SP
7337  015362  000114                            JMP     (R4)            ;RTS
7338
7339                                    ;SUBROUTINE TO PUSH 4 WORDS ONTO THE STACK
7340
7341  015364  012704  000510            PUSHR:  MOV     #STACK0,R4      ;SET R4 TO STACK
7342  015370  012524                            MOV     (R5)+,  (R4)+   ;PUT DATA ON STACK
7343  015372  012524                            MOV     (R5)+,  (R4)+   ;
```

```
7344  015374  012524                          MOV     (R5)+,  (R4)+   ;
7345  015376  012524                          MOV     (R5)+,  (R4)+   ;
7349  015400                                  MTPS    (R5)+           ;SET THE PROCESSOR STATUS
 (1)  015400  106425                          .WORD   106400!..C
7353  015402  005205                          INC     R5
7354  015404  012577  163054                  MOV     (R5)+,  @FISVEC ;SET UP FIS ERROR TRAP VECTOR
7355  015410  012577  163052                  MOV     (R5)+,  @FISLVL ;TRAP STATUS
7356  015414  000205                          RTS     R5              ;RETURN
7357
7358
7359                                  ;SUBROUTINE TO POP 2 WORDS OFF THE STACK
7360                                  ;ALSO SAVES THE PROCESSOR STATUS WORD (EXCEPT T BIT)
7361
7365  015416                          POPR:   MFPS    $PSW            ;SAVE PROCESSOR STATUS WORD
 (1)  015416  106767                          .WORD   106700!..C
7369  015422  042767  000020  163002          BIC     #20,    $PSW    ;CLEAR T-BIT
7370  015430  016767  163060  163000          MOV     STACK4, ANS1    ;SAVE THE ANSWER
7371  015436  016767  163054  162774          MOV     STACK6, ANS2    ;
7372  015444  000207                          RTS     PC
7373
7374
7375                                  ;SUBROUTNE TO POP 6 WORDS OFF THE STACKS.
7376                                  ;THE TWO OFF THE R6 STACK WERE PUT ON BY THE ERROR TRAP.
7377                                  ;THE FOUR OFF THE SOFTWARE STACK WERE THE ORIGINAL INPUT DATA.
7378                                  ;ALSO SAVES THE PS AND STACK POINTER AFTER THE FIS TRAP.
7379
7383  015446                          POPER:  MFPS    $PSW            ;SAVE PROCESSOR STATUS WORD
 (1)  015446  106767                          .WORD   106700!..C
7387  015452  012667  000056          POPER1: MOV     (SP)+,  SAVRTS  ;SAVE RTS ADDRESS
7388  015456  012667  162754                  MOV     (SP)+,  ANS1    ;SAVE RTI ADDRESS
7389  015462  011667  162752                  MOV     (SP),   ANS2    ;SAVE RTI STATUS
7390  015466  042767  000020  162744          BIC     #20,    ANS2    ;CLEAR THE T-BIT
7391  015474  012746  015502                  MOV     #1$,    -(SP)
7392  015500  000002                          RTI                     ;RESTORE PROCESSOR STATUS
7393  015502  016767  163002  162732  1$:     MOV     STACK0, ANS3    ;SAVE DATA
7394  015510  016767  162776  162726          MOV     STACK2, ANS4    ;
7395  015516  016767  162772  162722          MOV     STACK4, ANS5    ;
7396  015524  016767  162766  162716          MOV     STACK6, ANS6    ;
7397  015532  000137                          JMP     @(PC)+          ;SIMULATED RTS
7398  015534  000000                  SAVRTS: 0
7399
7400                                  ;SUBROUTINE TO PUSH 4 WORDS ONTO THE PC STACK
7401
7402  015536  012504                  PUSH7:  MOV     (R5)+,  R4      ;SET R4 TO STACK
7403  015540  012524                          MOV     (R5)+,  (R4)+   ;PUT DATA ON STACK
7404  015542  012524                          MOV     (R5)+,  (R4)+   ;
7405  015544  012524                          MOV     (R5)+,  (R4)+   ;
7406  015546  012524                          MOV     (R5)+,  (R4)+   ;
7410  015550                                  MTPS    (R5)+           ;SET THE PROCESSOR STATUS
 (1)  015550  106425                          .WORD   106400!..C
7414  015552  005205                          INC     R5
7415  015554  012577  162704                  MOV     (R5)+,  @FISVEC ;SET UP FIS ERROR TRAP VECTOR
7416  015560  012577  162702                  MOV     (R5)+,  @FISLVL ;TRAP STATUS
7417  015564  000205                          RTS     R5              ;RETURN
7418
7419                                  ;SUBROUTINE TO POP 4 WORDS OFF THE PC "STACK"
```

```
7420                                        ;ALSO SAVES THE PROCESSOR STATUS WORD (EXCEPT T BIT)
7421
7425  015566                                POP7:   MFPS    $PSW            ;SAVE PROCESSOR STATUS WORD
 (1)  015566  106767                              .WORD   106700!.^C
7429  015572  042767  000020  162632              BIC     #20,    $PSW    ;CLEAR T-BIT
7430  015600  011600                              MOV     (SP),   R0      ;GET RETURN ADDRESS
7431  015602  162700  000014                      SUB     #14,    R0      ;POINT TO TOP OF "PC STACK"
7432  015606  012067  162624                      MOV     (R0)+,  ANS1    ;SAVE 1ST HALF INPUT DATA
7433  015612  012067  162622                      MOV     (R0)+,  ANS2    ;SAVE 2ND HALF INPUT DATA
7434  015616  010067  162612                      MOV     R0,     $SP     ;SAVE ASSUMED END PC "STACK POINTER"
7435  015622  012067  162614                      MOV     (R0)+,  ANS3    ;SAVE 1ST HALF OF ANSWER
7436  015626  012067  162612                      MOV     (R0)+,  ANS4    ;SAVE 2ND HALF OF ANSWER
7437  015632  000207                              RTS     PC
7438
7439                                        ;ERRONIOUS TRAP SERVICE ROUTINE
7440
7441  015634  104000                        TRAPER: HLT                     ;FIS SHOULDN'T HAVE TRAPED
7442  015636  000523                              523                     ;THE ERROR NUMBER IS 523
7443  015640  000002                              RTI
7444
```

```
7445                                    ;*****************************************************************
 (1)                                    .SBTTL  HLT ROUTINE (ERROR TYPEOUT)
 (1)  015642  032737  002000  000422  HLT$:   BIT     #SW10,@#$SWREG  ;SHOULD IT RING THE BELL ON ERROR?
 (1)  015650  001402                        BEQ     1$              ;NO - SKIP
 (1)  015652  000004  000504                TYPE    ,$BELL          ;RING BELL
 (1)  015656  005267  162600          1$:   INC     ERRORS          ;COUNT THE NUMBER OF ERRORS
 (1)  015662  032737  020000  000422        BIT     #SW13,@#$SWREG  ;SKIP TYPEOUT IF SET
 (1)  015670  001023                        BNE     2$              ;SKIP TYPEOUTS
 (1)  015672  000004  000472                TYPE    ,RETURN
 (1)  015676  013637  000402                MOV     @(6)+,@#$FATAL  ;PLACE THE ERROR NUMBER IN LOCATION $FATAL
 (1)  015702  014667  162522                MOV     -(6),HLTAD$     ;PUT ADDRESS OF INSTRUCTION ON STACK
 (1)  015706  162767  000002  162514        SUB     #2,HLTAD$
 (2)  015714  017605  000000                MOV     @(6),TTY        ;TYPE @(6) IN OCTAL
 (2)  015720  004767  000124                JSR     %7,PRINTR       ;TYPE LEADING ZERO'S
 (1)  015724  062716  000002                ADD     #2,(6)          ;ADJUST THE RETURN ADDRESS
 (1)  015730  000004  000500                TYPE    ,SPACE+3
 (1)  015734  004767  000046                JSR     PC,ERROR$       ;GO TO USER ERROR ROUTINE
 (1)  015740  105767  162454          2$:   TSTB    $ENV            ;ARE WE RUNNING UNDER APT?
 (1)  015744  001403                        BEQ     4$              ;IF NOT THEN GO TO 4$
 (1)  015746  005237  000400                INC     @#$MSGTY        ;OTHERWISE INFORM APT
 (1)  015752  000777                        BR      .               ;AND LOOP
 (1)  015754  005737  000422          4$:   TST     @#$SWREG        ;HALT ON ERROR
 (1)  015760  100001                        BPL     .+4             ;SKIP IF CONTINUE
 (1)  015762  000000                        HALT                    ;HALT ON ERROR!
 (1)  015764  032737  001000  000422        BIT     #SW09,@#$SWREG  ;CHECK FOR INHIBIT LOOP ON ERROR
 (1)  015772  001001                        BNE     .+4             ;SKIP IF LOOP ON ERROR
 (1)  015774  000002                        RTI
 (1)  015776  105067  162501                CLRB    $ICNT
 (1)  016002  000167  177164                JMP     KIT$            ;LOOP ON TEST UNTIL NO ERRORS
 (1)
7446
```

```
7448
7449                                    ;******************************************************************
7450
7451                                    .SBTTL  USER ERROR ROUTINE
7452
7453  016006  117767  162416  162524 ERRORS: MOVB    @HLTAD$,TYPCNT  ;TYPE COUNT IS LOW BYTE OF HLT
7454  016014  062767  000002  162516         ADD     #2,     TYPCNT  ;TYPE COUNT = X+2
7455  016022  012703  000430                MOV     #HLTAD$,R3      ;TOP OF DATA TO BE TYPED
7456  016026                          ERR1$:
 (1)  016026  012305                        MOV     (R3)+,TTY       ;TYPE (R3)+ IN OCTAL
 (1)  016030  004767  000014                JSR     %7,PRINTR       ;TYPE LEADING ZERO'S
7457  016034  000004  000501                TYPE,   SPACE+4         ;SPACE
7458  016040  105367  162474                DECB    TYPCNT          ;CHECK FOR DONE
7459  016044  100370                        BPL     ERR1$           ;BRANCH IF NOT DONE
7460  016046  000207                        RTS     PC
7461
```

```
 7463  016050  112767  000001  162466  PRINTR: MOVB    #1,.PR          ;SET ZERO FILL SWITCH
   (1)  016056  000402                          BR      .+6             ;SKIP
   (1)  016060  005067  162460          PRINTS: CLR     .PR             ;SUPRESS LEADING ZERO'S
   (1)  016064  112767  177772  162453          MOVB    #-6,.PR+1       ;SET COUNT
   (1)  016072  010446                          MOV     R4,-(6)         ;SAVE R4
   (1)  016074  012704  016176                  MOV     #.PRBUF,R4      ;SET POINTER TO FIRST ASCII CHAR.
   (1)  016100  105014                          CLRB    (4)             ;CLEAR FIRST BYTE
   (1)  016102  000405                          BR      .PRF            ;ROTATE FIRST BIT
   (1)  016104  105014                  .PRL:   CLRB    (4)             ;CLEAR BYTE OF CHARACTER
   (1)  016106  006105                          ROL     TTY             ;ROTATE BIT INTO C
   (1)  016110  106114                          ROLB    (4)             ;PACK IT
   (1)  016112  006105                          ROL     TTY             ;ROTATE BIT INTO C
   (1)  016114  106114                          ROLB    (4)             ;PACK IT
   (1)  016116  006105                  .PRF:   ROL     TTY             ;ROTATE BIT INTO C
   (1)  016120  106114                          ROLB    (4)             ;PACK IT
   (1)  016122  105714                          TSTB    (4)             ;IS IT ZERO?
   (1)  016124  001402                          BEQ     .+6             ;SKIP INC
   (1)  016126  105267  162412                  INCB    .PR             ;SET FILL SWITCH
   (1)  016132  105767  162406                  TSTB    .PR             ;CHECK FILL SWITCH
   (1)  016136  001402                          BEQ     .+6             ;SKIP BITSET
   (1)  016140  152724  000060                  BISB    #'0,(4)+        ;MAKE INTO ASCII CHAR
   (1)  016144  105267  162375                  INCB    .PR+1           ;INC COUNT
   (1)  016150  001355                          BNE     .PRL            ;REPEAT
   (1)  016152  022704  016176                  CMP     #.PRBUF,R4      ;EMPTY BUFFER?
   (1)  016156  001002                          BNE     .+6             ;SKIP IF NOT
   (1)  016160  112724  000060                  MOVB    #'0,(4)+        ;LOAD 1 ZERO
   (1)  016164  105014                          CLRB    (4)             ;NULL TERMINATOR
   (1)  016166  000004  016176                  TYPE    .PRBUF          ;TYPE IT
   (1)  016172  012604                          MOV     (6)+,R4         ;RESTORE R4
   (1)  016174  000207                          RTS     PC              ;RETURN
   (1)
   (1)  016176  000004                  .PRBUF: .BLKW   4               ;OUTPUT BUFFER
```

```
 7465                                   ;*****************************************************************
  (1)
  (1)                                   .SBTTL  POWER DOWN AND UP ROUTINES
  (1)
  (1)                                   ;POWER DOWN ROUTINE
  (1)  016206  012737  016330  000024   $PWRDN: MOV     #$ILLUP,@#PWRVEC ;;SET FOR FAST UP
  (1)  016214  012737  000340  000026          MOV     #340,@#PWRVEC+2 ;;PRIO:7
  (3)  016222  010046                        MOV     R0,-(SP)        ;;PUSH R0 ON STACK
  (3)  016224  010146                        MOV     R1,-(SP)        ;;PUSH R1 ON STACK
  (3)  016226  010246                        MOV     R2,-(SP)        ;;PUSH R2 ON STACK
  (3)  016230  010346                        MOV     R3,-(SP)        ;;PUSH R3 ON STACK
  (3)  016232  010446                        MOV     R4,-(SP)        ;;PUSH R4 ON STACK
  (3)  016234  010546                        MOV     R5,-(SP)        ;;PUSH R5 ON STACK
  (1)  016236  010667  000072                MOV     SP,$SAVR6       ;;SAVE SP
  (1)  016242  012737  016254  000024        MOV     #$PWRUP,@#PWRVEC ;;SET UP VECTOR
  (1)  016250  000000                        HALT
  (1)  016252  000776                        BR      .-2             ;;HANG UP
  (1)
  (1)                                   ;POWER UP ROUTINE
  (1)  016254  016706  000054           $PWRUP: MOV     $SAVR6,SP       ;;GET SP
  (1)  016260  005067  000050                CLR     $SAVR6          ;;WAIT LOOP FOR THE TTY
  (1)  016264  005267  000044           1$:     INC     $SAVR6          ;;WAIT FOR THE INC
  (1)  016270  001375                        BNE     1$              ;;OF  WORD
  (3)  016272  012605                        MOV     (SP)+,R5        ;;POP STACK INTO R5
  (3)  016274  012604                        MOV     (SP)+,R4        ;;POP STACK INTO R4
  (3)  016276  012603                        MOV     (SP)+,R3        ;;POP STACK INTO R3
  (3)  016300  012602                        MOV     (SP)+,R2        ;;POP STACK INTO R2
  (3)  016302  012601                        MOV     (SP)+,R1        ;;POP STACK INTO R1
  (3)  016304  012600                        MOV     (SP)+,R0        ;;POP STACK INTO R0
  (1)  016306  012737  016206  000024        MOV     #$PWRDN,@#PWRVEC ;;SET UP THE POWER DOWN VECTOR
  (1)  016314  012737  000340  000026        MOV     #340,@#PWRVEC+2 ;;PRIO:7
  (1)  016322  000004                        TYPE                    ;REPORT THE POWER FAILURE
  (1)  016324  016336                $PWRMG: .WORD   $POWER          ;;POWER FAIL MESSAGE POINTER
  (1)  016326  000002                        RTI
  (1)  016330  000000                $ILLUP: HALT                    ;;THE POWER UP SEQUENCE WAS STARTED
  (1)  016332  000776                        BR      .-2             ;;  BEFORE THE POWER DOWN WAS COMPLETE
  (1)  016334  000000                $SAVR6: 0                       ;;PUT THE SP HERE
  (1)  016336  005015  047520  042527 $POWER: .ASCIZ  <15><12>"POWER"
  (1)  016344  000122
  (1)                                       .EVEN
```

```
7470
7471                                    ;*      TYPE OUT ROUTINE
7472                                    ;*      ----------------
7473                                    ;*
7474                                    ;*
7475                                    ;*      THIS ROUTINE IS USED TO TYPE ASCIZ MESSAGES
7476                                    ;*
7477
7478  016346  132737  000040  000421  $TYPE:  BITB    #40,@#$ENVM     ;HAS THE CONSOLE OUTPUTS BEEN SUPPRESSED?
7479  016354  001007                    BNE     3$              ;IF SO THEN RETURN FROM THE SUBROUTINE VIA 3$
7480  016356  010046                    MOV     R0,-(SP)        ;OTHERWISE SAVE R0
7481  016360  017600  000002            MOV     @2(SP),R0       ;GET THE ADDRESS OF THE ASSCIZ STRING
7482  016364  112046            2$:     MOVB    (R0)+,-(SP)     ;PUSH THE CHARACTER TO BE TYPED ONTO STACK
7483  016366  001005                    BNE     4$              ;BRANCH IF IT IS NOT THE TERMINATOR
7484  016370  005726                    TST     (SP)+
7485  016372  012600                    MOV     (SP)+,R0        ;OTHERWISE RESTORE THE STACK AND R0
7486  016374  062716  000002    3$:     ADD     #2,(SP)         ;ADJUST THE RETURN PC
7487  016400  000002                    RTI                     ;AND RETURN
7488
7489  016402  105777  162142    4$:     TSTB    @$TPS           ;IS THE PRINTER AVAILABLE?
7490  016406  100375                    BPL     4$              ;IF NOT THEN LOOP HERE
7491  016410  112677  162136            MOVB    (SP)+,@$TPB     ;OUT PUT THE CHARACTER
7492  016414  000763                    BR      2$              ;AND GO BACK
7493          000001            .END
```

DVKACA.SRC CROSS REFERENCE TABLE -- USER SYMBOLS

```
A      = 015074          7261#   7266
ABASE  = 000000          5339
ACDW1  = 000000          5339
ACDW2  = 000000          5339
ACPUOP= 000000          5339
ADDW0  = 000000          5339
ADDW1  = 000000          5339
ADDW10= 000000          5339
ADDW11= 000000          5339
ADDW12= 000000          5339
ADDW13= 000000          5339
ADDW14= 000000          5339
ADDW15= 000000          5339
ADDW2  = 000000          5339
ADDW3  = 000000          5339
ADDW4  = 000000          5339
ADDW5  = 000000          5339
ADDW6  = 000000          5339
ADDW7  = 000000          5339
ADDW8  = 000000          5339
ADDW9  = 000000          5339
ADEVCT= 000000          5339
ADEVM  = 000000          5339
AENV   = 000000          5339
AENVM  = 000000          5339
AFATAL= 000000          5339
AMADR1= 000000          5339
AMADR2= 000000          5339
AMADR3= 000000          5339
AMADR4= 000000          5339
AMAMS1= 000000          5339
AMAMS2= 000000          5339
AMAMS3= 000000          5339
AMAMS4= 000000          5339
AMSGAD= 000000          5339
AMSGLG= 000000          5339
AMSGTY= 000000          5339
AMTYP1= 000000          5339
AMTYP2= 000000          5339
AMTYP3= 000000          5339
AMTYP4= 000000          5339
ANS1    000436          5348#   5349    7147    7148    7149    7150    7151    7152    7153    7154    7155    7156    7157
                        7158    7159    7163    7164    7168    7169    7173    7174    7175    7176    7177    7178    7179
                        7180    7181    7185    7189    7193    7194    7195    7196    7197    7198    7200    7202    7206
                        7210    7214    7215    7216    7217    7219    7223    7227    7231    7232    7238*   7243    7244
                        7251    7252    7308*   7327*   7370*   7388*   7432*
ANS2    000440          5350#   5351    7147    7148    7149    7150    7151    7152    7153    7154    7155    7156    7157
                        7158    7159    7163    7164    7168    7169    7173    7174    7175    7176    7177    7178    7179
                        7180    7181    7185    7189    7193    7194    7195    7196    7197    7198    7200    7202    7206
                        7210    7214    7215    7216    7217    7219    7223    7227    7231    7232    7238*   7243    7244
                        7309*   7328*   7329*   7371*   7389*   7390*   7433*
ANS3    000442          5352#   5353    7159    7163    7164    7168    7169    7185    7189    7200    7202    7206    7210
                        7219    7223    7227    7231    7232    7243    7251    7252    7332*   7393*   7435*
ANS4    000444          5354#   7159    7163    7164    7168    7169    7185    7189    7200    7202    7206    7210    7219
                        7223    7227    7231    7232    7243    7251    7252    7333*   7394*   7436*
ANS5    000446          5355#   7163    7164    7168    7169    7185    7189    7206    7210    7223    7227    7231    7232
```

```
                    7243    7251    7252    7334*   7395*
ANS6    000450     5356*   7163    7164    7168    7169    7185    7189    7206    7210    7223    7227    7231    7232
                    7243    7251    7252    7335*   7396*
APASS = 000000     5339
APRIOR= 000000     5339
ASWREG= 000000     5339
ATESTN= 000000     5339
AUNIT = 000000     5339
AUSWR = 000000     5339
AVECT1= 000000     5339
AVECT2= 000000     5339
BEGIN   000600     7116    7122*   7150    7151    7156    7157    7164    7169    7175    7177    7185    7197    7210
                   7215    7232    7251    7252    7258
ENDCT   015022     7264*
END1    001040     7147*
END10   002252     7154*
END11   002410     7155*
END12   002542     7156*
END13   002676     7157*
END14   003036     7158*
END15   003210     7159*
END16   003442     7163*
END17   003672     7164*
END2    001174     7148*
END20   004130     7168*
END21   004362     7169*
END22   004522     7173*
END23   004656     7174*
END24   005012     7175*
END25   005150     7176*
END26   005304     7177*
END27   005444     7178*
END3    001330     7149*
END30   005604     7179*
END31   005744     7180*
END32   006102     7181*
END33   006334     7185*
END34   006572     7189*
END35   006726     7193*
END36   007066     7194*
END37   007222     7195*
END4    001462     7150*
END40   007356     7196*
END41   007512     7197*
END42   007650     7198*
END43   010024     7200*
END44   010200     7202*
END45   010434     7206*
END46   010666     7210*
END47   011024     7214*
END5    001614     7151*
END50   011160     7215*
END51   011316     7216*
END52   011452     7217*
END53   011626     7219*
END54   012064     7223*
```

```
END55   012322          7227#
END56   012560          7231#
END57   013012          7232#
END6    001754          7152#
END60   013230          7238#
END61   013506          7243#
END62   013704          7244#
END63   014330          7251#
END64   014740          7252#
END7    002114          7153#
ERRORS  000462          5358#   7445*
ERROR$  016006          7445    7453#
ERR1$   016026          7456#   7459
FISLVL  000466          5360#   7292*   7355*   7416*
FISVEC  000464          5359#   7291*   7354*   7415*
HLT   = 104000          5296#   7147    7148    7149    7150    7151    7152    7153    7154    7155    7156    7157    7158
                        7159    7163    7164    7168    7169    7173    7174    7175    7176    7177    7178    7179    7180
                        7181    7185    7189    7193    7194    7195    7196    7197    7198    7200    7202    7206    7210
                        7214    7215    7216    7217    7219    7223    7227    7231    7232    7238    7243    7244    7251
                        7252    7441
HLTAD$  000430          5342#   5343    7445*   7453    7455
HLT$    015642          7128    7445#
ISA17   003550          7164#
ISA21   004236          7169#
ISA33   006210          7185#
ISA46   010542          7210#
ISA57   012666          7232#
ISR16   003276          7163#
ISR17   003524          7164#
ISR20   003760          7168#
ISR21   004212          7169#
ISR33   006164          7185#
ISR34   006422          7189#
ISR45   010266          7206#
ISR46   010516          7210#
ISR54   011714          7223#
ISR55   012152          7227#
ISR56   012410          7231#
ISR57   012642          7232#
ISR61   013336          7243#
ISR62   013614          7244#
ISR63   014050          7251#
ISR64   014462          7252#
KIT$    015172          7268#   7445
LAD$    000470          5361#   7140*   7251*   7252*   7268*
N     = 000065          5317#   7147#   7148#   7149#   7150#   7151#   7152#   7153#   7154#   7155#   7156#   7157#   7158#
                        7159#   7163#   7164#   7168#   7169#   7173#   7174#   7175#   7176#   7177#   7178#   7179#   7180#
                        7181#   7185#   7189#   7193#   7194#   7195#   7196#   7197#   7198#   7200#   7202#   7206#   7210#
                        7214#   7215#   7216#   7217#   7219#   7223#   7227#   7231#   7232#   7237    7238#   7243#   7244#
                        7251#   7252#
OVERS   015176          7268#
PC    =%000007          5305#   7147*   7148*   7149*   7150*   7151*   7152*   7153*   7154*   7155*   7156*   7157*   7158*
                        7159*   7163*   7164*   7168*   7169*   7173*   7174*   7175*   7176*   7177*   7178*   7179*   7180*
                        7181*   7185*   7189*   7193*   7194*   7195*   7196*   7197*   7198*   7200*   7202*   7206*   7210*
                        7214*   7215*   7216*   7217*   7219*   7223*   7227*   7231*   7232*   7243*   7244*   7251*   7252*
                        7258*   7372*   7397    7437*   7445*   7460*   7463*
```

```
POPER   015446          7163  7168  7189  7206  7223  7227  7231  7243  7244  7251  7252  7383#
POPER1  015452          7251  7252  7387#
POPES   015304          7164  7169  7185  7210  7232  7322#
POPR    015416          7147  7148  7149  7152  7153  7154  7155  7158  7163  7168  7173  7174  7176
                        7178  7179  7180  7181  7189  7193  7194  7195  7196  7198  7206  7214  7216
                        7217  7223  7227  7231  7243  7365#
POPS    015252          7150  7151  7156  7157  7164  7169  7175  7177  7185  7197  7210  7215  7232
                        7302#
POP7    015566          7159  7200  7202  7219  7425#
PRINTR  016050          7445  7456  7463#
PRINTS  016060          7463#
PUSHR   015364          7147  7148  7149  7152  7153  7154  7155  7158  7163  7168  7173  7174  7176
                        7178  7179  7180  7181  7189  7193  7194  7195  7196  7198  7206  7214  7216
                        7217  7223  7227  7231  7243  7244  7251  7252  7341#
PUSHS   015212          7150  7151  7156  7157  7164  7169  7175  7177  7185  7197  7210  7215  7232
                        7276#
PUSH7   015536          7159  7200  7202  7219  7402#
PWRVEC= 000024          5306#  7465*
RESTRT  000222          7116#
RETURN  000472          5362#  7251  7252  7445
RTA16   003260          7163#
RTA17   003506          7164#
RTA20   003742          7168#
RTA21   004174          7169#
RTA33   006146          7185#
RTA34   006404          7189#
RTA45   010250          7206#
RTA46   010500          7210#
RTA54   011676          7223#
RTA55   012134          7227#
RTA56   012372          7231#
RTA57   012624          7232#
RTA61   013320          7243#
RTA62   013576          7244#
RTA63   014042          7251#
RTA64   014454          7252#
R0      =%000000        5297#  7111*  7112*  7113  7127*  7135*  7136*  7137*  7138*  7147*  7174*  7179*  7206*
                        7214*  7252*  7258*  7430*  7431*  7432  7433  7434  7435  7436  7465*  7480  7481*
                        7482  7485*
R1      =%000001        5298#  7148*  7153*  7168*  7173*  7198*  7223*  7251*  7465*
R2      =%000002        5299#  7149*  7181*  7194*  7216*  7243*  7465*
R3      =%000003        5300#  7163*  7178*  7189*  7196*  7217*  7455*  7456  7465*
R4      =%000004        5301#  7155*  7158*  7176*  7193*  7227*  7237*  7238*  7251*  7252*  7254  7255*  7307*
                        7311  7326*  7337  7341*  7342*  7343*  7344*  7345*  7402*  7403*  7404*  7405*  7406*
                        7463*  7465*
R5      =%000005        5302#  7147*  7148*  7149*  7150*  7151*  7152*  7153*  7154*  7155*  7156*  7157*  7158*
                        7159*  7163*  7164*  7168*  7169*  7173*  7174*  7175*  7176*  7177*  7178*  7179*  7180*
                        7181*  7185*  7189*  7193*  7194*  7195*  7196*  7197*  7198*  7200*  7202*  7206*  7210*
                        7214*  7215*  7216*  7217*  7219*  7223*  7227*  7231*  7232*  7243*  7244*  7251*  7252*
                        7277*  7278  7279  7280  7281  7282*  7286*  7290*  7291  7292  7293  7342  7343
                        7344  7345  7349*  7353*  7354  7355  7356*  7402  7403  7404  7405  7406  7410*
                        7414*  7415  7416  7417*  7465*
SAVRTS  015534          7387*  7398#
SAVTPS  000506          5367#
SCOPE = 104400          5307#  7147  7148  7149  7150  7151  7152  7153  7154  7155  7156  7157  7158
                        7159  7163  7164  7168  7169  7173  7174  7175  7176  7177  7178  7179  7180
```

```
                    7181    7185    7189    7193    7194    7195    7196    7197    7198    7200    7202    7206    7210
                    7214    7215    7216    7217    7219    7223    7227    7231    7232    7237    7243    7244    7251
                    7252    7258
SCOPES$  015074     7130    7268#
SP    =%000006      5304#   7122*   7150*   7151*   7156*   7157*   7164*   7169*   7175*   7177*   7185*   7197*   7210*
                    7215*   7232*   7251*   7252*   7276    7278*   7279*   7280*   7281*   7307    7308    7309    7310
                    7326    7327    7328    7330*   7332    7333    7334    7335    7336    7387    7388    7389    7391*
                    7430    7465*   7480*   7481    7482*   7484    7485    7486*   7491
SPACE    000475     5363#   7445    7457
STACK0   000510     5368#   5374    7147    7148    7149    7152    7153    7154    7155    7158    7163    7168    7173
                    7174    7176    7178    7179    7180    7181    7189    7193    7194    7195    7196    7198    7206
                    7214    7216    7217    7223    7227    7231    7251    7252    7341    7393
STACK1=  000511     5374#
STACK2   000512     5369#   7394
STACK4   000514     5370#   7147    7148    7149    7152    7153    7154    7155    7158    7173    7174    7176    7178
                    7179    7180    7181    7193    7194    7195    7196    7198    7214    7216    7217    7370    7395
STACK6   000516     5371#   7371    7396
STACK8   000520     5372#
STAK10   000532     5373#   7237    7238
STK15    003104     7159#
STK43    007716     7200#
STK44    010072     7202#
STK53    011520     7219#
SVLAD$   015160     7268#
SW08  =  000400     5283    5315#   7268
SW09  =  001000     5283    5314#   7445
SW10  =  002000     5283    5313#   7445
SW11  =  004000     5283    5312#   7268
SW12  =  010000     5283    5311#
SW13  =  020000     5283    5310#   7445
SW14  =  040000     5283    5309#   7268
SW15  =  100000     5283    5308#
TEMP     000534     5375#   7251*   7252*
TIMES    000536     5376#   7110*   7250*   7253*   7268
TRAPER   015634     7147    7148    7149    7150    7151    7152    7153    7154    7155    7156    7157    7158    7159
                    7173    7174    7175    7176    7177    7178    7179    7180    7181    7193    7194    7195    7196
                    7197    7198    7200    7202    7214    7215    7216    7217    7219    7243    7244    7251    7252
                    7441#
TSA12    002500     7156#
TSA13    002632     7157#
TSA24    004746     7175#
TSA26    005240     7177#
TSA4     001420     7150#
TSA41    007446     7197#
TSA5     001552     7151#
TSA50    011114     7215#
TST1     000722     7147#
TST10    002132     7154#
TST11    002270     7155#
TST12    002426     7156#
TST13    002560     7157#
TST14    002714     7158#
TST15    003054     7159#
TST16    003226     7163#
TST17    003460     7164#
TST2     001056     7148#
```

```
TST20   003710          7168#
TST21   004146          7169#
TST22   004400          7173#
TST23   004540          7174#
TST24   004674          7175#
TST25   005030          7176#
TST26   005166          7177#
TST27   005322          7178#
TST3    001212          7149#
TST30   005462          7179#
TST31   005622          7180#
TST32   005762          7181#
TST33   006120          7185#
TST34   006352          7189#
TST35   006610          7193#
TST36   006744          7194#
TST37   007104          7195#
TST4    001346          7150#
TST40   007240          7196#
TST41   007374          7197#
TST42   007530          7198#
TST43   007666          7200#
TST44   010042          7202#
TST45   010216          7206#
TST46   010452          7210#
TST47   010704          7214#
TST5    001500          7151#
TST50   011042          7215#
TST51   011176          7216#
TST52   011334          7217#
TST53   011470          7219#
TST54   011644          7223#
TST55   012102          7227#
TST56   012340          7231#
TST57   012576          7232#
TST6    001632          7152#
TST60   013030          7237#
TST61   013246          7243#
TST62   013524          7244#
TST63   013742          7251#
TST64   014354          7252#
TST7    001772          7153#
TTY    =000005          5303#   7445*   7456*   7463*
TTYOUT  000546          5380#   7251    7252    7254*
TYPCNT  000540          5377#   7453*   7454*   7458*
TYPE   =000004          5316#   7251    7252    7258    7445    7457    7463    7465
YESRT   000542          5378#   7123
$APTHD  000430          5340#   5341
$BELL   000504          5366#   7445
$CPUOP  000426          5339#
$DEVCT  000410          5339#   7111
$DOAGN  015052          7258#
$ENDAD  015042          5330    7258#
$ENDCT  015022          7258#   7262
$ENDMG  015056          7258#
$ENULL  015070          7258#
```

```
$ENV    000420          5339# 7133  7445
$ENVM   000421          5339# 7251  7252  7478
$EOP    014776          7258#
$EOPCT  015014          7258#
$ETABL  000420          5339#
$ETEND  000430          5339# 5340
$F    = 000524          5318# 7147# 7148# 7149# 7150# 7151# 7152# 7153# 7154# 7155# 7156# 7157# 7158#
                        7159# 7163# 7164# 7168# 7169# 7173# 7174# 7175# 7176# 7177# 7178# 7179# 7180#
                        7181# 7185# 7189# 7193# 7194# 7195# 7196# 7197# 7198# 7200# 7202# 7206# 7210#
                        7214# 7215# 7216# 7217# 7219# 7223# 7227# 7231# 7232# 7238# 7243# 7244# 7251#
                        7252# 7442#
$FATAL  000402          5339# 7445*
$GET42  015032          7258#
$HD   = 000003          5057
$HIBTS  000430          5340#
$ICNT   000503          5364# 7268* 7445*
$ILLUP  016330          7465#
$MAIL   000400          5339# 5340  7113
$MBADR  000432          5340#
$MSGAD  000414          5339#
$MSGLG  000416          5339#
$MSGTY  000400          5339# 7445*
$PASS   000406          5339# 7258*
$PASTM  000436          5340#
$POWER  016336          7465#
$PSW    000432          5344# 5345  7147  7148  7149  7150  7151  7152  7153  7154  7155  7156  7157
                        7158  7159  7163  7164  7168  7169  7173  7174  7175  7176  7177  7178  7179
                        7180  7181  7185  7189  7193  7194  7195  7196  7197  7198  7200  7202  7206
                        7210  7214  7215  7216  7217  7219  7223  7227  7231  7232  7238* 7243  7244*
                        7251  7252  7302* 7306* 7322* 7365* 7369* 7383* 7425* 7429*
$PWRDN  016206          7125  7465#
$PWRMG  016324          7465#
$PWRUP  016254          7465#
$SAVR6  016334          7465#*
$SETUP= 000020          5328# 7258
$SP     000434          5346# 5347  7147* 7148* 7149* 7152* 7153* 7154* 7155* 7158* 7163* 7168* 7173*
                        7174* 7176* 7178* 7179* 7180* 7181* 7189* 7193* 7194* 7195* 7196* 7198* 7206*
                        7214* 7216* 7217* 7223* 7227* 7231* 7238* 7243* 7244* 7251* 7252* 7310* 7336*
                        7434*
$STUP = 177777          5328#
$SVPC = 001000          5330#
$SWR  = 160000          5057# 7258  7465
$SWREG  000422          5339# 7268  7445
$TESTN  000404          5339# 7139* 7147  7148  7149  7150  7151  7152  7153  7154  7155  7156  7157
                        7158  7159  7163  7164  7168  7169  7173  7174  7175  7176  7177  7178  7179
                        7180  7181  7185  7189  7193  7194  7195  7196  7197  7198  7200  7202  7206
                        7210  7214  7215  7216  7217  7219  7223  7227  7231  7232  7238  7243  7244
                        7251  7252  7268*
$TN   = 000001          5057#
$TPB    000552          5382# 7135  7251* 7252* 7491*
$TPS    000550          5381# 7251* 7252* 7489
$TSTM   000434          5340#
$TYPE   016346          7124  7478#
$UNIT   000412          5339#
$UNITM  000440          5340#
$USWR   000424          5339#
```

D08

DVKACA.SRC CROSS REFERENCE TABLE -- USER SYMBOLS

```
.      = 016416          5321#  5326   5330#  5338#  5340#  5341#  5343#  5345#  5347#  5349#  5351#  5353#  7109#
                         7120#  7147   7148   7149   7150   7151   7152   7153   7154   7155   7156   7157   7158
                         7159   7163   7164   7168   7169   7173   7174   7175   7176   7177   7178   7179   7180
                         7181   7185   7189   7193   7194   7195   7196   7197   7198   7200   7202   7206   7210
                         7214   7215   7216   7217   7219   7223   7227   7231   7232   7237#  7238#  7243#  7244#
                         7251#  7252#  7258#  7261   7262#  7266#  7286#  7302#  7322#  7349#  7365#  7383#  7410#
                         7425#  7445   7463#  7465
.BIT   = 177777          5051#  5283   7445
.PR      000544          5379#  7463#
.PRBUF   016176          7463#
.PRF     016116          7463#
.PRL     016104          7463#
.$X    = 000430          5340#
..A    = 015566          7237#  7238#  7243#  7244#  7251#  7252#  7286#  7302#  7322#  7349#  7365#  7383#  7410#
                         7425#
..B    = 015572          7237#  7238#  7243#  7244#  7251#  7252#  7286#  7302#  7322#  7349#  7365#  7383#  7410#
                         7425#
..C    = 000067          7237#  7238#  7243#  7244#  7251#  7252#  7286#  7302#  7322#  7349#  7365#  7383#  7410#
                         7425#
```

```
COMMEN    974#
DUMP     5132#  7445    7456
ENDCOM    986#
ESCAPE   1097#
MFPS     5070#  7238    7244    7302    7322    7365    7383    7425
MTPS     5085#  7237    7243    7244    7251    7252    7286    7349    7410
MULT     3466#
NEWTST   1030#
POP      1483#  5056#   7465
PRINT    5142#
PUSH     1475#  5056#   7465
REPORT   4437#
SDUMP    5137#
SETUP     801#
SKIP     1131#
SLASH     926#
STARS     895#  5055#   5319    5330    5339    5340    7104    7258    7268    7445    7449    7465
TYPBIN   1419#
TYPDEC   1389#
TYPNAM   1177#
TYPNUM   1356#
TYPOCS   1309#
TYPOCT   1272#
TYPTXT   1226#
$ADDER   5605#  7163    7168
$ADDES   5507#  7164    7169
$ADDR    5443#  7147    7148    7149    7152    7153    7154    7155    7158
$ADDS    5383#  7150    7151    7156    7157
$ADRER   6110#  7243    7244
$ALL     6881#  7237
$ALL2    6914#  7238
$DIVER   6779#  7223    7227    7231
$DIVES   6681#  7232
$DIVR    6617#  7214    7216    7217
$DIVS    6553#  7215
$FATL    5059#  7147    7148    7149    7150    7151    7152    7153    7154    7155    7156    7157    7158    7159    7163
         7164   7168    7169    7173    7174    7175    7176    7177    7178    7179    7180    7181    7185    7189    7193
         7194   7195    7196    7197    7198    7200    7202    7206    7210    7214    7215    7216    7217    7219    7223
         7227   7231    7232    7238    7243    7244    7251    7252    7442
$FINT    6972#  7251    7252
$FIS7    6031#  7159    7200    7202    7219
$HLT     5181#  7445
$MULER   6451#  7206
$MULES   6353#  7210
$MULR    6289#  7193    7194    7195    7196    7198
$MULS    6233#  7197
$OCTAL   5226#  7463
$SCOPE   5150#  7268
$SUBER   5929#  7189
$SUBES   5831#  7185
$SUBR    5767#  7173    7174    7176    7178    7179    7180    7181
$SUBS    5707#  7175    7177
$SWDOC   5099#  5283
$$ESCA   1110#
$$FATL   5066#  7147    7148    7149    7150    7151    7152    7153    7154    7155    7156    7157    7158    7159    7163
         7164   7168    7169    7173    7174    7175    7176    7177    7178    7179    7180    7181    7185    7189    7193
```

DVKACA.SRC CROSS REFERENCE TABLE -- MACRO NAMES

```
          7194    7195    7196    7197    7198    7200    7202    7206    7210    7214    7215    7216    7217    7219    7223
          7227    7231    7232    7238    7243    7244    7251    7252    7442
$$NEWT    1064#
$$SKIP    1144#
.EQUAT     168#
.HEADE      39#    5055#   5057
.KT11      308#
.SETUP     747#    5055#   5328
.SWRHI      81#
.$ACTI    4032#    5055#   5330
.$APTB    4080#    5055#   5339
.$APTH    4350#    5055#   5340
.$APTY    4513#
.$ASTA    4397#
.$CATC     485#
.$CMTA     586#
.$DB2D    3666#
.$DB2O    3790#
.$DIV     3568#
.$EOP     1535#    5055#   7258
.$ERRO    1947#
.$ERRT    2140#
.$MULT    3504#
.$POWE    3223#    5056#   7465
.$RAND    3286#
.$RDDE    2919#
.$RDOC    2827#
.$READ    2613#
.$R2AZ    3934#
.$SAVE    2995#
.$SB2D    3751#
.$SB2O    3853#
.$SCOP    1739#
.$SIZE    3348#
.$SUPR    3891#
.$TRAP    3095#
.$TYPB    2528#
.$TYPD    2450#
.$TYPE    2228#
.$TYPO    2353#
.$40CA     515#
```

DVKACA.SRC CROSS REFERENCE TABLE -- PERMANENT SYMBOLS

```
ADD     7277  7282  7445  7454  7486
BEQ     7134  7147  7148  7149  7150  7151  7152  7153  7154  7155  7156  7157  7158  7159  7163
        7164  7168  7169  7173  7174  7175  7176  7177  7178  7179  7180  7181  7185  7189  7193
        7194  7195  7196  7197  7198  7200  7202  7206  7210  7214  7215  7216  7217  7219  7223
        7227  7231  7232  7238  7243  7244  7251  7252  7258  7268  7445  7463
BGT     7258
BIC     7238  7258  7306  7329  7369  7390  7429
BISB    7463
BIT     7268  7445
BITB    7133  7251  7252  7478
BNE     7114  7251  7252  7268  7445  7463  7465  7479  7483
BPL     7251  7252  7445  7459  7490
BR      7150  7151  7156  7157  7163  7164  7168  7169  7175  7177  7185  7189  7197  7206  7210
        7215  7223  7227  7231  7232  7243  7244  7251  7252  7445  7463  7465  7492
CLR     7112  7139  7140  7237  7238  7243  7244  7246  7251  7252  7255  7286  7302  7322  7349
        7365  7383  7410  7425  7463  7465
CLRB    7251  7252  7445  7463
CMP     7113  7147  7148  7149  7150  7151  7152  7153  7154  7155  7156  7157  7158  7159  7163
        7164  7168  7169  7173  7174  7175  7176  7177  7178  7179  7180  7181  7185  7189  7193
        7194  7195  7196  7197  7198  7200  7202  7206  7210  7214  7215  7216  7217  7219  7223
        7227  7231  7232  7238  7243  7244  7251  7252  7463
CMPB    7147  7148  7149  7150  7151  7152  7153  7154  7155  7156  7157  7158  7159  7163  7164
        7168  7169  7173  7174  7175  7176  7177  7178  7179  7180  7181  7185  7189  7193  7194
        7195  7196  7197  7198  7200  7202  7206  7210  7214  7215  7216  7217  7219  7223  7227
        7231  7232  7238  7243  7244  7251  7252  7268
DEC     7251  7252  7258
DECB    7458
EMT     5296
FADD    7147  7148  7149  7150  7151  7152  7153  7154  7155  7156  7157  7158  7159  7163  7164
        7168  7169  7238  7243  7251
FDIV    7214  7215  7216  7217  7219  7223  7227  7231  7232  7238
FMUL    7193  7194  7195  7196  7197  7198  7202  7206  7210  7238  7244  7252
FSUB    7173  7174  7175  7176  7177  7178  7179  7180  7181  7185  7189  7200  7237
HALT    5326  7445  7465
INC     7258  7290  7353  7414  7445  7465
INCB    7268  7463
IOT     5316
JMP     7116  7258  7293  7311  7337  7397  7445
JSR     7147  7148  7149  7150  7151  7152  7153  7154  7155  7156  7157  7158  7159  7163  7164
        7168  7169  7173  7174  7175  7176  7177  7178  7179  7180  7181  7185  7189  7193  7194
        7195  7196  7197  7198  7200  7202  7206  7210  7214  7215  7216  7217  7219  7223  7227
        7231  7232  7243  7244  7251  7252  7258  7445  7456
MOV     7110  7111  7122  7123  7124  7125  7127  7128  7129  7130  7131  7132  7135  7136  7137
        7138  7147  7148  7149  7150  7151  7152  7153  7154  7155  7156  7157  7158  7163  7164
        7168  7169  7173  7174  7175  7176  7177  7178  7179  7180  7181  7185  7189  7193  7194
        7195  7196  7197  7198  7206  7210  7214  7215  7216  7217  7223  7227  7231  7232  7237
        7238  7243  7244  7245  7250  7251  7252  7253  7254  7258  7268  7278  7279  7280  7281
        7291  7292  7307  7308  7309  7310  7326  7327  7328  7330  7332  7333  7334  7335  7336
        7341  7342  7343  7344  7345  7354  7355  7370  7371  7387  7388  7389  7391  7393  7394
        7395  7396  7402  7403  7404  7405  7406  7415  7416  7430  7432  7433  7434  7435  7436
        7445  7455  7456  7463  7465  7480  7481  7485
MOVB    7251  7252  7268  7453  7463  7482  7491
NOP     7147  7148  7149  7150  7151  7152  7153  7154  7155  7156  7157  7158  7159  7163  7164
        7168  7169  7173  7174  7175  7176  7177  7178  7179  7180  7181  7185  7189  7193  7194
        7195  7196  7197  7198  7200  7202  7206  7210  7214  7215  7216  7217  7219  7223  7227
        7231  7232  7237  7243  7244  7258
```

DVKACA.SRC CROSS REFERENCE TABLE -- PERMANENT SYMBOLS

```
RESET   7258
ROL     7463
ROLB    7463
RTI     7251  7252  7268  7331  7392  7443  7445  7465  7487
RTS     7356  7372  7417  7437  7460  7463
RTT     5378
SUB     7431  7445
TRAP    5307
TST     7147  7148  7149  7150  7151  7154  7155  7156  7159  7163  7164  7174  7176  7193  7195
        7196  7198  7200  7202  7206  7216  7217  7219  7251  7252  7268  7276  7445  7484
TSTB    7148  7159  7163  7174  7181  7214  7251  7252  7268  7445  7463  7489
.ABS    5049
.ASCII  7258
.ASCIZ  5362  5363  7465
.BLKW   7463
.BYTE   5339  5364  7258
.ENABL     4
.END    7493
.ENDC   5057  5283  5319  5328  5330  5339  5340  7104  7147  7148  7149  7150  7151  7152  7153
        7154  7155  7156  7157  7158  7159  7163  7164  7168  7169  7173  7174  7175  7176  7177
        7178  7179  7180  7181  7185  7189  7193  7194  7195  7196  7197  7198  7200  7202  7206
        7210  7214  7215  7216  7217  7219  7223  7227  7231  7232  7238  7243  7244  7251  7252
        7258  7268  7445  7449  7463  7465
.EVEN   5339  5365  7258  7465
.IF     5057  5283  5319  5328  5330  5339  5340  7104  7147  7148  7149  7150  7151  7152  7153
        7154  7155  7156  7157  7158  7159  7163  7164  7168  7169  7173  7174  7175  7176  7177
        7178  7179  7180  7181  7185  7189  7193  7194  7195  7196  7197  7198  7200  7202  7206
        7210  7214  7215  7216  7217  7219  7223  7227  7231  7232  7238  7243  7244  7251  7252
        7258  7268  7445  7449  7463  7465
.IFF    5319  5330  5339  5340  7104  7147  7148  7149  7150  7151  7152  7153  7154  7155  7156
        7157  7158  7159  7163  7164  7168  7169  7173  7174  7175  7176  7177  7178  7179  7180
        7181  7185  7189  7193  7194  7195  7196  7197  7198  7200  7202  7206  7210  7214  7215
        7216  7217  7219  7223  7227  7231  7232  7238  7243  7244  7251  7252  7258  7268  7445
        7449  7465
.IFNE   7243  7244
.IIF    5057  5339  7258  7463
.IRP    5328  7465
.LIST      2  4623  5053  5054  5281  5327  5328  5335  5339  7145  7147  7148  7149  7150  7151
        7152  7153  7154  7155  7156  7157  7158  7159  7162  7163  7164  7167  7168  7169  7172
        7173  7174  7175  7176  7177  7178  7179  7180  7181  7184  7185  7188  7189  7192  7193
        7194  7195  7196  7197  7198  7200  7202  7205  7206  7209  7210  7213  7214  7215  7216
        7217  7219  7222  7223  7226  7227  7230  7231  7232  7236  7237  7238  7241  7243  7244
        7249  7251  7252  7258  7263  7267  7271  7284  7285  7286  7289  7300  7301  7302  7305
        7320  7321  7322  7325  7347  7348  7349  7352  7363  7364  7365  7368  7381  7382  7383
        7386  7408  7409  7410  7413  7423  7424  7425  7428  7442  7463  7468
.MACRO     39    81   166   308   485   515   586   747   801   895   926   974   986  1030  1064
        1097  1110  1131  1144  1177  1226  1272  1309  1356  1389  1419  1475  1483  1535  1739
        1947  2140  2228  2353  2450  2528  2613  2827  2919  2995  3095  3223  3286  3348  3466
        3504  3568  3666  3751  3790  3853  3891  3934  4032  4080  4350  4397  4437  4513  5059
        5066  5070  5085  5099  5132  5137  5142  5150  5181  5226  5383  5443  5507  5605  5707
        5767  5831  5929  6031  6110  6233  6289  6353  6451  6553  6617  6681  6779  6881  6914
        6972
.MCALL  5055  5056
.MEXIT  5339
.NLIST     1     3  5050  5052  5279  5322  5328  5333  5339  7143  7147  7148  7149  7150  7151
        7152  7153  7154  7155  7156  7157  7158  7159  7160  7163  7164  7165  7168  7169  7170
```

```
        7173    7174    7175    7176    7177    7178    7179    7180    7181    7182    7185    7186    7189    7190    7193
        7194    7195    7196    7197    7198    7200    7202    7203    7206    7207    7210    7211    7214    7215    7216
        7217    7219    7220    7223    7224    7227    7228    7231    7232    7234    7237    7238    7239    7243    7244
        7247    7251    7252    7258    7260    7265    7269    7283    7286    7287    7288    7299    7302    7303    7304
        7319    7322    7323    7324    7346    7349    7350    7351    7362    7365    7366    7367    7380    7383    7384
        7385    7407    7410    7411    7412    7422    7425    7426    7427    7442    7463    7466
.NTYPE  7237    7238    7243    7244    7251    7252    7286    7302    7322    7349    7365    7383    7410    7425
.PAGE   5045    5047    5336    6232    7103    7146    7147    7148    7149    7150    7151    7152    7153    7154    7155
        7156    7157    7158    7163    7164    7168    7169    7173    7174    7175    7176    7177    7178    7179    7180
        7181    7185    7189    7193    7194    7195    7196    7197    7198    7199    7201    7206    7210    7214    7215
        7216    7217    7218    7223    7227    7231    7232    7233    7242    7243    7244    7251    7252    7257    7268
        7272    7445    7447    7462    7464    7469
.REPT   4624    5323
.SBTTL  5280    5330    5334    5339    5340    7106    7144    7161    7166    7171    7183    7187    7191    7204    7208
        7212    7221    7225    7229    7235    7240    7248    7258    7268    7270    7445    7451    7463    7465    7467
.TITLE  5057
.WORD   5330    5339    5340    5366    7147    7148    7149    7150    7151    7152    7153    7154    7155    7156    7157
        7158    7159    7163    7164    7168    7169    7173    7174    7175    7176    7177    7178    7179    7180    7181
        7185    7189    7193    7194    7195    7196    7197    7198    7200    7202    7206    7210    7214    7215    7216
        7217    7219    7223    7227    7231    7232    7237    7238    7243    7244    7251    7252    7258    7286    7302
        7322    7349    7365    7383    7410    7425    7465


ERRORS DETECTED:  0
DEFAULT GLOBALS GENERATED:  0

*,DVKACA/CRF=DVKACA.SML,DVKACA.SRC
RUN-TIME: 44 56 5 SECONDS
RUN-TIME RATIO: 568/106=5.3
CORE USED:  41K  (81 PAGES)
```

JOB

Spooler runtime 12 Seconds, 52 KCS, 254 disk reads, 1 disk writes, 99 pages