LSI-11

FIS INSTRUCTION

CVKACC0

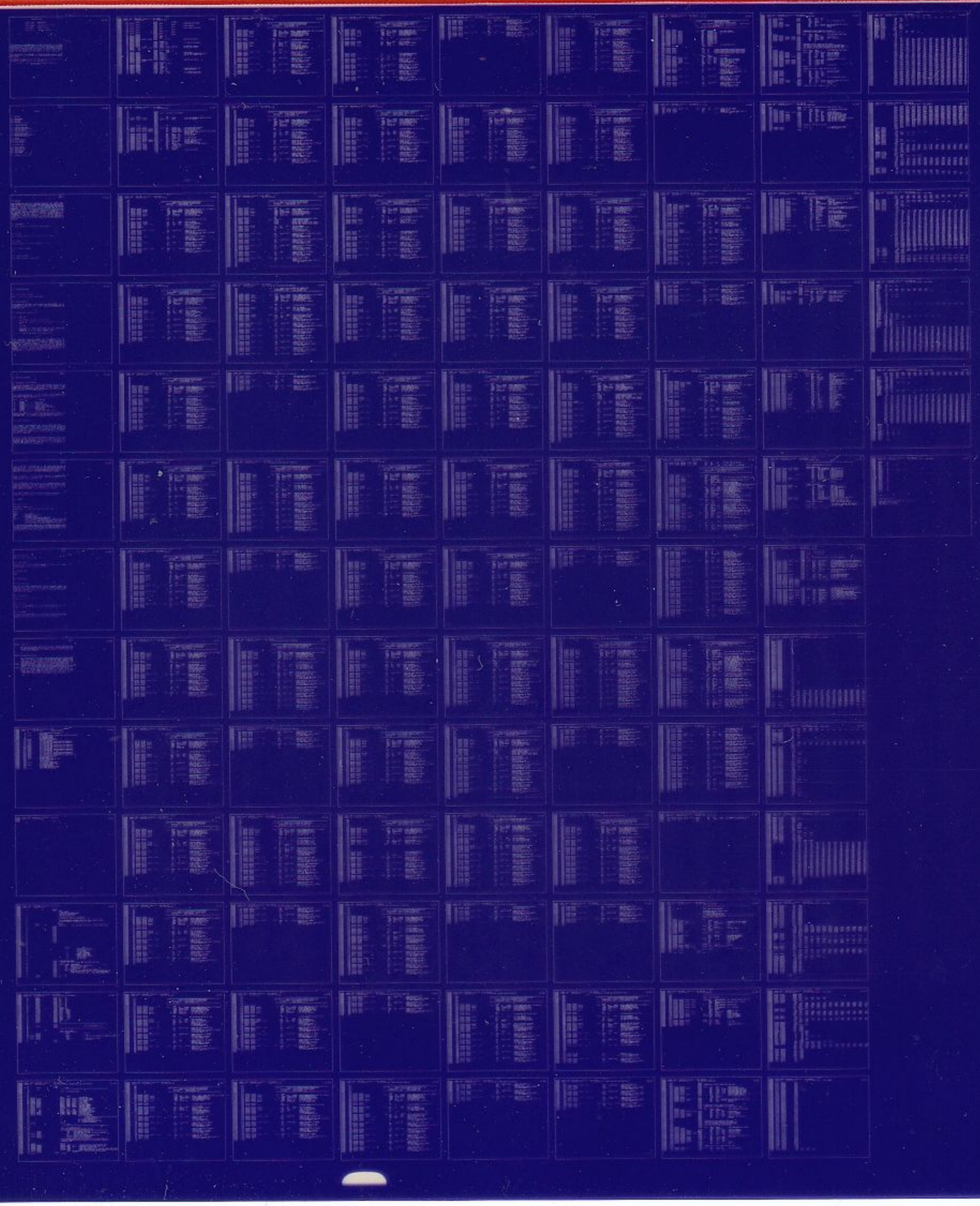
AH-8196C-MC

FICHE 1 OF 1

JAN 1979

digital

MADE IN USA



IDENTIFICATION

PRODUCT CODE: AC-8194C-MC

PRODUCT NAME: CVKACCO LSI-11 FIS INST

DATE CREATED: AUGUST 22,1975

MAINTAINER: DIAGNOSTIC GROUP

CONTENTS

1.0 ABSTRACT

2.0 REQUIREMENTS
2.1 Equipment
2.2 Storage
2.3 Preliminary Programs

3.0 LOADING PROCEDURE

4.0 STARTING PROCEDURE
4.1 Control Switch Settings
4.2 Starting Address
4.3 Program And/or Operator Action

5.0 OPERATING PROCEDURE
5.1 Operational Switch Settings
5.2 Subroutine Abstracts

6.0 ERRORS
6.1 Error Printout
6.2 Error Recovery
6.3 Error Counter

7.0 RESTRICTIONS

8.0 MISCELLANEOUS
8.1 Execution Time
8.2 Stack Pointer
8.3 Pass Counter
8.4 Test Number
8.5 Power Fail

9.0 PROGRAM DESCRIPTION

1.0 ABSTRACT

This program tests the LSI-11 floating instruction set <FADD, FSUB, FMUL, and FDIV> option with fixed number patterns, using each register at least once as the stack pointer. It also checks stack overflow and that the floating instructions can be interrupted (by the console teletype). [However, this test will not be executed when bit 5 of $ENVM byte is high]. The program should be run for at least 2 passes with all switches low. The program is designed to run under APT, and ACT, systems. When running under APT, with bit 5 of $ENVM low it will be required to have a SLU with TTY registers having addresses of 176560-66 and interrupt vectors of 70 for receiver and 74 for transmitter. Under such a condition it will also be required to change the run time of first pass from 5 seconds to the time given in Sec. 8.1, and the run time for the longest test from 3 seconds to 30 seconds.

2.0 REQUIREMENTS

LTC SWITCH MUST BE IN OFF POSITION TO RUN DIAGNOSTIC.

2.1 Equipment

LSI-11 standard computer with FIS option and 4K of memory.

2.2 Storage

Program Storage - The routines use memory 0 - 17500.

2.3 Preliminary Programs

None.

3.0 LOADING PROCEDURE

Use standard procedure for ABS Tapes.

4.0 STARTING PROCEDURE

4.1 Control Switch Settings

See 5.1.1 (all low for worst case testing).

4.2 Starting Address

After loading the program it should always be started at 200. If it is desired to save the pass counter then the program should be restarted at location RESTRT (i.e. 222) otherwise the program can be restarted at 200.

4.3 Program And/or Operator Action

4.3.1 Stand Alone -

1. Load program into memory using ABS loader.
2. Set switches (see Sec 5.1.1) all low except bit 7 for worst case.
3. Type 200G.
4. The program will loop and 'END PASS'' will be typed after completion of every pass. However type out will be suppressed if bit 5 of location $ENVM is high.
5. A minimum of two passes should always be run.

4.3.2 Under Apt - Load the program, set the switches (see Sec. 5.1.1) and start. When under APT, with bit 5 of $ENVM low it will be required to have a SLU with TTY registers having addresses of 176560-66 and interrupt vectors of 70 for receiver and 74 for transmitter. Under such a condition it will also be required to change the run time of first pass from 5 seconds to the time given in Sec. 8.1, and the run time for the longest test from 3 seconds to 30 seconds. The test times and pass times are suggested with bit 7 of $SWREG, low, if it is desired to enable the iterations then the times should be multiplied by a factor of 256.

5.0 OPERATING PROCEDURE

5.1 Operational Switch Settings

All switches low except SW<11> is worst case testing. With bit 11 of the location $SWREG (i.e. location 422), high each subtest will be looped upon until completion of 256 passes of that subtest. 'END PASS' will be typed upon completion of a pass of the entire program. Alternate pass will run with the T-bit set.

5.1.1 Switch Settings Are - A 16 bit location called $SWREG (i.e. location 422) has been used to give the following options by inserting a 1 in their respective positions.

BIT #	OCTAL VALUE	FUNCTION
15	100000.....	Halt On Error
14	040000.....	Scope Loop
13	020000.....	Inhibit Printout
12	010000.....	Inhibit Trace Trapping
11	004000.....	Enable Iterations Of Subtest
10	002000.....	Bell On Error
09	001000.....	Loop On Error
08	0004XX.....	Loop On Test In Bits 7 Thru 0

An 8 bit byte $ENVM (i.e. location 421) has been used to define the operating mode. All typeouts can be suppressed by making bit 5 of byte $ENVM high, in other words by placing a 20000 in location 420.

5.2 Subroutine Abstracts

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

5.2.2 HLT - This routine (called by an EMT instruction) prints out an error message (see 6.1). If SW<9> is a 1 and a HLT is executed, the subtest will be looped upon until 256 consecutive good passes are completed. To inhibit typeouts, make SW<13> a 1. To ring the bell on an error, make SW<10> a 1. A high bit 5 in location $ENVM will inhibit any typeouts and ringing of bells.

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

5.2.4 Trap Catcher - A ".+2" - "HALT" sequence is repeated from 0-776 to catch any unexpected traps. Thus any unexpected traps or interrupts will halt at the vector +2.

5.2.5 Floating Error Trap (To 244) - If a floating point error (overflow, underflow, or divide by zero) was expected, the vector will point to a unique ISR within the subtest where the error occurred which checks the data on the stack(s). If an error was not anticipated, an erroneous trap will be detected in traper.

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

6.0 ERRORS

6.1 Error Printout

The format is as follows:

ERRNM ADR PS SP ANS1 ANS2 ANS3 ANS4 ANS5 ANS6

Where:

ERRNM	:	= Error Number
ADR		= Address of Error HLT
PS		= Processor Status
SP		= Contents of Stack Pointer Register
ANS1-6		= Error Data Read from the Stack(s). From 0 to 6 of these may be typed depending on the number following the HLT; e.g., HLT+3 would type ANS1 thru ANS3, HLT (by itself) would stop after ERRNM, ADR, PS, and SP.

To find the failing test, loop at the listing above the address typed. In most cases the comment beside the HLT tells what was being checked and what was expected. All printouts will be suppressed when bit 5 of location $ENVM is high. While running under APT the diagnostic will not support spooling of console outputs.

6.2 Error Recovery

Restart at 200 or 222 (see Sec. 4.2).

6.3 Error Counter

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

7.0 RESTRICTIONS

None.

8.0 MISCELLANEOUS

8.1 Execution Time

Due to the random characteristics of the interrupt tests, the execution time can be half a minute or more. However, normally "END PASS" will be typed within 40 seconds with all switches down. Execution time will increase by a factor of 256 when iterations of each subtest are enabled.

8.2 Stack Pointer

Stack is initially set to 600.

8.3 Pass Count

A 16 bit location "$PASS" (i.e. location 406) is used to keep pass count. It can be cleared by restarting the program at 200.

8.4 Test Number

A 16 bit location "$TESTN" (i.e. locaton 404) is used to keep track of the test number. Upper byte of this location gives the iteration number and the lower byte the test that was being executed.

00008.5 Power Fail

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

00009.0 PROGRAM DESCRIPTION

This program tests all the FIS instructions of the LSI-11 (FADD, FSUB, FMUL, and FDIV). All registers are checked to see if they function properly as the stack pointer. The program has many subtests (the code between 2 scope statements) which are run once before continuing to the next subtest. SW<11> set to a 1 causes each subtest to be run
00256 times SW<9> set to a 1 enables loop on error. The location $ICNT contains the iteration count and the location $TESTN contains the test number. All the subtests should be run sequentially by starting at
00200 not by starting at the beginning of the subtest. To loop on a particular subtest, put the test number (see listing) in the right byte of the location $SWREG and SW<8> set to a 1. This test will be looped upon until SW<8> is set to a 0 or the right byte is changed. If the test is non-existent, the program will be run as usual.

.ENDR

CVKACC MACY11 30A(1052) 21-AUG-78 15:28
CVKACC.P11 16-AUG-78 08:41 TABLE OF CONTENTS SEQ 0009

00100			
00200			
00300			
00400	32	55100	SWITCH OPTIONS AND ASSIGNMENTS
00500	81		ACT11 HOOKS
00600	91	60500	VECTOR AREA, STACKS, ANSWER AREA, AND SETUP ROUTINE
00700	96		APT MAILBOX-ETABLE
00800	126		APT PARAMETER BLOCK
00900	193	38300	STARTING OF THE PROGRAM
01000	226	41700	FADD TEST SECTION
01100	879	43500	TEST FLOATING ADD INSTRUCTION WITH UNDERFLOW
01200	1024	44000	TEST FLOATING ADD INSTRUCTION WITH OVERFLOW
01300	1169	44500	FSUB TEST SECTION
01400	1740	46000	TEST FLOATING SUB. INSTRUCTION WITH UNDERFLOW
01500	1813	46400	TEST FLOATING SUB. INSTRUCTION WITH OVERFLOW
01600	1886	46800	FMUL TEST SECTION
01700	2271	48100	TEST FLOATING MUL. INSTRUCTION WITH UNDERFLOW
01800	2344	48500	TEST FLOATING MUL. INSTRUCTION WITH OVERFLOW
01900	2417	48900	FDIV TEST SECTION
02000	2656	49800	TEST FLOATING DIV. INSTRUCTION WITH UNDERFLOW
02100	2729	50200	TEST FLOATING DIV INSTRUCTION WITH OVERFLOW
02200	2802	50600	TEST FLOATING DIV. INSTRUCTION FOR DIVIDE BY ZERO
02300	2953	51200	TEST OF ALL FIS AT ONCE
02400	3016	51700	ADDRESS ERROR TEST
02500	3145	52500	INTERUPT ABORT TEST SECTION
02600	3369		END OF PASS ROUTINE
02700	3408		SCOPE ROUTINE
02800	3433	54700	PUSH AND POP SUBROUTINES
02900	3567		HLT ROUTINE (ERROR TYPEOUT)
03000	3600	72800	USER ERROR ROUTINE
03100	3613		OCTAL WORD & ADDRESS TYPER
03200	3648		POWER DOWN AND UP ROUTINES
03300	3686	74400	ASCIZ TYPE OUT ROUTINE

```
00100    CVKACC  MACY11 30A(1052)  21-AUG-78  15:28  PAGE 2                    SEQ 0010
00200    CVKACC.P11    16-AUG-78 08:41
00300
00400         1                                31700
```

```
                                31900
                                32000   .ABS
                                        .TITLE  CVKACC
                                        ;*COPYRIGHT (C) AUGUST 1978
                                        ;*DIGITAL EQUIPMENT CORP.
                                        ;*MAYNARD, MASS. 01754
                                        ;*
                                        ;*PROGRAM BY DIAGNOSTIC ENGINERRING
                                        ;*
                                        ;*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
                                32900
                                33600
                                34000
                                35400
                                35500
                                36900
                                40200
                                40700
                                41200
                                42000
                                45100
                                49600
                                54900
                                55300


                                        ;       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

                                55500   ;ERROR MESSAGE FORMAT:
                                55600   ;  ERRNM   ADR  PSW  SP  ANS1 ANS2 ANS3 ANS4 ANS5 ANS6
                                55700   ;
                                55800   ;WHERE  ERRNM=  ERROR NUMBER
                                55900   ;       ADR =   ADDRESS OF ''HLT'' INSTRUCTION + 2
                                56000   ;       PSW =   PROCESSOR STATUS WORD
                                56100   ;       SP =    STACK POINTER
                                56200   ;       ANS1 THRU ANS6 =        DATA OFF THE STACK(S)
                                56300   ;       NOTE:   ANS1 THRU ANS6 ARE NOT ALWAYS TYPED, DEPENDING ON THE
                                56400   ;               NUMBER ADDED TO THE 'HLT''. ''HLT'' ALONE TYPES NONE,
                                56500   ;               ''HLT+1'' TYPES ANS1, ''HLT+2'' TYPES ANS1 AND ANS2, ETC.
                                56600
104000                          56700   HLT=    EMT
000000                          56800   R0=     %0
```

```
00300
00400      58          000001                  56900   R1=      %1
00500      59          000002                  57000   R2=      %2
00600      60          000003                  57100   R3=      %3
00700      61          000004                  57200   R4=      %4
00800      62          000005                  57300   R5=      %5
00900      63          000005                  57400   TTY=     %5
01000      64          000006                  57500   SP=      %6
01100      65          000007                  57600   PC=      %7
01200      66          000024                  57700   PWRVEC=  24
01300      67          104400                  57800   SCOPE=   TRAP
01400      68          100000                  57900   SW15=    100000
01500      69          040000                  58000   SW14=     40000
01600      70          020000                  58100   SW13=     20000
01700      71          010000                  58200   SW12=     10000
01800      72          004000                  58300   SW11=      4000
01900      73          002000                  58400   SW10=      2000
02000      74          001000                  58500   SW09=      1000
02100      75          000400                  58600   SW08=       400
02200      76          000004                  58700   TYPE=    IOT
02300      77          000001                  58800   N=       1
02400      78          000001                  58900   $F=      1
02500      79                                          ;*****************************************************************
02600      80                                  59100
02700      81          000000                  59200   .=       0                      ;TRAP CATCHER FROM 0 - 776
02800      82                                  60000
02900      83                                          ;*****************************************************************
03000      84
03100      85                                          .SBTTL  ACT11 HOOKS
03200      86                                          ;HOOKS REQUIRED BY ACT11
03300      87          001000                                  $SVPC=.                ;SAVE PC
03400      88          000046                                  .=46
03500      89  000046  015664                                  $ENDAD                 ;;1)SET LOC.46 TO ADDRESS OF $ENDAD IN .$EOP
03600      90          000052                                  .=52
03700      91  000052  000000                                  .WORD   0              ;;2)SET LOC.52 TO ZERO
03800      92          001000                                  .=$SVPC                ;; RESTORE PC
03900      93                                  60200
04000      94                                  60300
```

```
                                    60800
                000400              60900   .=      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 VER
        000440  000000                      $UNITM: .WORD               ;;ADDITIONAL RUN TIME (SECS) OF A PASS FOR EACH ADDI
        000442  000014                              .WORD   $ETEND-$MAIL/2 ;;LENGTH MAILBOX-ETABLE(WORDS)
                000430              61200           .=      $APTHD
        000430                      61300   HLTAD$:
```

```
00100
00200
00300
00400     151            000432                    61400                .=      HLTAD$+2
00500     152  000432                               61500  $PSW:
00600     153            000434                    61600                .=      $PSW+2
00700     154  000434                               61700  $SP:                  ;PROCESSOR STATUS WORD
00800     155            000436                    61800                .=      $SP+2
00900     156  000436                               61900  ANS1:                 ;STACK POINTER
01000     157            000440                    62000                .=      ANS1+2
01100     158  000440                               62100  ANS2:                 ;FIRST ANSWER (SEE CODE)
01200     159            000442                    62200                .=      ANS2+2
01300     160  000442                               62300  ANS3:
01400     161            000444                    62400                .=      ANS3+2
01500     162  000444  000000                      62500  ANS4:   0
01600     163  000446  000000                      62600  ANS5:   0
01700     164  000450  000000                      62700  ANS6:   0
01800     165  000452  000000  000000  000000      62800          0,0,0,0       ;NON-%6 STACK BUFFER
01900     166  000460  000000
02000     167  000462  000000                      62900  ERRORS: 0
02100     168  000464  000244                      63000  FISVEC: 244           ;FIS TRAP VECTOR ADDRESS
02200     169  000466  000246                      63100  FISLVL: 246
02300     170  000470  000000                      63200  LAD$:   0
02400     171  000472  006412      000             63300  RETURN: .ASCIZ  <12><15>      ;RETURN AND LINEFEED
02500     172  000475     015  020012  020040      63400  SPACE:  .ASCIZ  <15><12>''  ''  ;RETURN AND 3 SPACES
02600     173  000502     000
02700     174  000503     000                      63500  $ICNT:  .BYTE   0
02800     175                                      63600          .EVEN
02900     176  000504  000007                      63700  $BELL:  .WORD   7             ;RING A BELL
03000     177  000506  000000                      63800  SAVTPS: 0                     ;LOC TO SAVE TELEPRINTER STATUS
03100     178  000510  000000                      63900  STACK0: 0                     ;NON-%6 STACK NORMAL LIMIT
03200     179  000512  000000                      64000  STACK2: 0                     ;
03300     180  000514  000000                      64100  STACK4: 0                     ;
03400     181  000516  000000                      64200  STACK6: 0                     ;
03500     182  000520  000000  000000  000000      64300  STACK8: 0,0,0,0,0             ;NON-%6 STACK BUFFER
03600     183  000526  000000  000000
03700     184  000532  000000                      64400  STAK10: 0
03800     185            000511                    64500  STACK1 = STACK0+1
03900     186  000534  000000                      64600  TEMP:   0
04000     187  000536  000000                      64700  TIMES:  0
04100     188  000540  000000                      64800  TYPCNT: 0
04200     189  000542  000006                      64900  YESRT:  RTT                   ;RETURN FROM TRACE TRAP
04300     190  000544  000000                      65000  .PR:    0                     ;COUNT AND SWITCH
04400     191  000546  000064                      65100  TTYOUT: 64
04500     192  000550  177564                      65200  $TPS:   177564                ;TTY PRINTER STATUS REG.
04600     193  000552  177566                      65300  $TPB:   177566                ;TTY PRINTER BUFFER REG.
```

```
00300
00400     194                                                 ;*****************************************************************
00500     195                                          38200
00600     196                                          38300  .SBTTL  STARTING OF THE PROGRAM
00700     197                                          38400
00800     198                                          38500
00900     199          000200                          38600  .=      200
01000     200 000200   000167  017104                 38700         JMP     NOOOP           ;GO TO TYPE HEADING & INITIALIZATION
01100     201 000204   000167  000370                 38800  RESTRT: JMP     BEGIN           ;START THE PROGRAM
01200     202                                          38900
01300     203                                          39000
01400     204                                          39100
01500     205                                          39200
01600     206          000600                          39300  .=      600
01700     207                                          39400
01800     208 000600   012706  000600                 39500  BEGIN:  MOV     #BEGIN, SP      ;INITIALIZE STACK POINTER
01900     209 000604   012737  000542  000014         39600         MOV     #YESRT, @#14    ;SET TRACE TRAP VECTOR
02000     210 000612   012737  017170  000020         39700         MOV     #$TYPE, @#20    ;SET UP VECTOR 20
02100     211 000620   012737  017030  000024         39800         MOV     #$PWRDN ,@#24   ;SERVICE POWER DOWN ROUTINE FOR ANY FUTURE
02200     212                                          39900                                 ;POWER DOWN
02300     213 000626   012700  000030                 40000         MOV     #30,    R0      ;SET R0 TO VECTOR 30
02400     214 000632   012720  016464                 40100         MOV     #HLT$,  (0)+    ;SET EMT VECTOR
02500     215 000636   012720  000340                 40200         MOV     #340,   (0)+
02600     216 000642   012720  015716                 40300         MOV     #SCOPE$,(0)+    ;SET TRAP VECTOR
02700     217 000646   012710  000340                 40400         MOV     #340,   (0)
02800     218 000652   012737  000006  000004         40500  1$:     MOV     #6,     @#4     ;RESTORE TIME-OUT VECTOR
02900     219 000660   132737  000001  000420         40600         BITB    #1,@#$ENV       ;ARE WE UNDER APT ?
03000     220 000666   001410                         40700         BEQ     2$              ;IF NOT THEN GO TO 2$
03100     221 000670   012700  000554                 40800         MOV     #$TPB+2,R0      ;OTHERWISE SET FOR THE OTHER SLU
03200     222 000674   012740  176566                 40900         MOV     #176566,-(R0)
03300     223 000700   012740  176564                 41000         MOV     #176564,-(R0)
03400     224 000704   012740  000074                 41100         MOV     #74,-(R0)
03500     225 000710   005067  177470                 41200  2$:     CLR     $TESTN
03600     226 000714   005067  177550                 41300         CLR     LAD$            ;CLEAR LOOP ADDRESS
03700     227                                          41400
03800     228                                          41500
```

```
00300
00400     229
00500     230
00600     231                                   ;*********************************************************************
00700     232                                   ;TEST 1:      FADD (LSI-11 FLOATING ADD INSTRUCTION)
00800     233                                   ;       000000,000000 + 000000,000000 = 000000,000000
00900     234                                   ;       PS = 004,       STACK POINTER = R0
01000     235                                   ;*********************************************************************
01100     236  000720  104400                              SCOPE
01200     237  000722  004567  015260           TST1:   JSR     R5,     PUSHR   ;PUSH 4 WORDS ONTO R0 STACK, SET PRIORITY
01300     238  000726  000000  000000                   .WORD   000000,000000   ;SECOND OPERAND ON TOP
01400     239  000732  000000  000000                   .WORD   000000,000000   ;FIRST OPERAND ON BOTTOM
01500     240  000736  000000                         .WORD   000             ;PROCESSOR PRIORITY LEVEL
01600     241  000740  016456  000340                 .WORD   TRAPER,340      ;FIS TRAP VECTOR
01700     242  000744  012700  000510                 MOV     #STACK0,R0      ;SET UP STACK POINTER
01800     243
01900     244  000750  000240                         NOP
02000     245  000752  075000                         FADD    R0              ;FLOATING ADD ON THE R0 STACK
02100     246
02200     247  000754  004767  015260                 JSR     PC,     POPR    ;POP THE ANSWER
02300     248  000760  010067  177450                 MOV     R0,     $SP     ;SAVE ''STACK POINTER''
02400     249  000764  122767  000004  177440       CMPB    #004,   $PSW    ;CHECK PS (EXCEPT T BIT)
02500     250  000772  001402                         BEQ     .+6             ;BRANCH IF OK
02600     251  000774  104000                         HLT                     ;PS NOT EQUAL TO 004
02700     252  000776  000001                         1                       ;THE ERROR NUMBER IS 1
02800     253
02900     254  001000  022767  000514  177426       CMP     #STACK4,$SP     ;CHECK THE STACK POINTER (R0)
03000     255  001006  001402                         BEQ     .+6             ;BRANCH IF OK
03100     256  001010  104000                         HLT                     ;STACK POINTER (R0) NOT EQUAL TO #STACK4
03200     257  001012  000002                         2                       ;THE ERROR NUMBER IS 2
03300     258
03400     259  001014  005767  177416                 TST     ANS1            ;CHECK FIRST HALF OF ANSWER
03500     260  001020  001402                         BEQ     .+6             ;BRANCH IF OK
03600     261  001022  104002                         HLT+2                   ;ANS1 NOT EQUAL TO 000000
03700     262  001024  000003                         3                       ;THE ERROR NUMBER IS 3
03800     263
03900     264  001026  005767  177406                 TST     ANS2            ;CHECK SECOND HALF OF ANSWER
04000     265  001032  001402                         BEQ     .+6             ;BRANCH IF OK
04100     266  001034  104002                         HLT+2                   ;ANS2 NOT EQUAL TO 000000
04200     267  001036  000004                         4                       ;THE ERROR NUMBER IS 4
04300     268
04400     269  001040  122767  000001  177336 END1:  CMPB    #1,     $TESTN  ;CHECK THE TEST NUMBER
04500     270  001046  001402                         BEQ     .+6             ;BRANCH IF OK
04600     271  001050  104000                         HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
04700     272  001052  000005                         5                       ;THE ERROR NUMBER IS 5
04800     273
04900     274
```

```
00300
00400      275
00500      276                    ;*****************************************************************************
00600      277                    ;TEST 2:        FADD (LSI-11 FLOATING ADD INSTRUCTION)
00700      278                    ;       000000,000000 + 152525,052524 = 152525,052524
00800      279                    ;       PS = 010,       STACK POINTER = R3
00900      280                    ;*****************************************************************************
01000      281
01100      282  001054  104400                        SCOPE
01200      283  001056  004567  015124          TST2: JSR     R5,     PUSHR   ;PUSH 4 WORDS ONTO R3 STACK, SET PRIORITY
01300      284  001062  152525  052524                .WORD   152525,052524   ;SECOND OPERAND ON TOP
01400      285  001066  000000  000000                .WORD   000000,000000   ;FIRST OPERAND ON BOTTOM
01500      286  001072  000040                        .WORD   040             ;PROCESSOR PRIORITY LEVEL
01600      287  001074  016456  000340                .WORD   TRAPER,340      ;FIS TRAP VECTOR
01700      288  001100  012703  000510                MOV     #STACK0,R3      ;SET UP STACK POINTER
01800      289
01900      290  001104  000240                        NOP
02000      291  001106  075003                        FADD    R3              ;FLOATING ADD ON THE R3 STACK
02100      292
02200      293  001110  004767  015124                JSR     PC,     POPR    ;POP THE ANSWER
02300      294  001114  010367  177314                MOV     R3,     $SP     ;SAVE ''STACK POINTER''
02400      295  001120  122767  000010  177304        CMPB    #010,   $PSW    ;CHECK PS (EXCEPT T BIT)
02500      296  001126  001402                        BEQ     .+6             ;BRANCH IF OK
02600      297  001130  104000                        HLT                     ;PS NOT EQUAL TO 010
02700      298  001132  000006                        6                       ;THE ERROR NUMBER IS 6
02800      299
02900      300  001134  022767  000514  177272        CMP     #STACK4,$SP     ;CHECK THE STACK POINTER (R3)
03000      301  001142  001402                        BEQ     .+6             ;BRANCH IF OK
03100      302  001144  104000                        HLT                     ;STACK POINTER (R3) NOT EQUAL TO #STACK4
03200      303  001146  000007                        7                       ;THE ERROR NUMBER IS 7
03300      304
03400      305  001150  022767  152525  177260        CMP     #152525,ANS1    ;CHECK FIRST HALF OF ANSWER
03500      306  001156  001402                        BEQ     .+6             ;BRANCH IF OK
03600      307  001160  104002                        HLT+2                   ;ANS1 NOT EQUAL TO 152525
03700      308  001162  000010                        10                      ;THE ERROR NUMBER IS 10
03800      309
03900      310  001164  022767  052524  177246        CMP     #052524,ANS2    ;CHECK SECOND HALF OF ANSWER
04000      311  001172  001402                        BEQ     .+6             ;BRANCH IF OK
04100      312  001174  104002                        HLT+2                   ;ANS2 NOT EQUAL TO 052524
04200      313  001176  000011                        11                      ;THE ERROR NUMBER IS 11
04300      314
04400      315  001200  122767  000002  177176  END2: CMPB    #2,     $TESTN  ;CHECK THE TEST NUMBER
04500      316  001206  001402                        BEQ     .+6             ;BRANCH IF OK
04600      317  001210  104000                        HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
04700      318  001212  000012                        12                      ;THE ERROR NUMBER IS 12
04800      319
04900      320
```

```
00100
00200
00300
00400      321
00500      322                ;*****************************************************************************
00600      323                ;TEST 3:        FADD (LSI-11 FLOATING ADD INSTRUCTION)
00700      324                ;       040200,000000 + 040200,000000 = 040400,000000
00800      325                ;       PS = 000,       STACK POINTER = R1
00900      326                ;*****************************************************************************
01000      327
01100      328  001214  104400                      SCOPE
01200      329  001216  004567  014764    TST3:     JSR     R5,     PUSHR   ;PUSH 4 WORDS ONTO R1 STACK, SET PRIORITY
01300      330  001222  040200  000000              .WORD   040200,000000   ;SECOND OPERAND ON TOP
01400      331  001226  040200  000000              .WORD   040200,000000   ;FIRST OPERAND ON BOTTOM
01500      332  001232  000040                      .WORD   040             ;PROCESSOR PRIORITY LEVEL
01600      333  001234  016456  000340              .WORD   TRAPER,340      ;FIS TRAP VECTOR
01700      334  001240  012701  000510              MOV     #STACK0,R1      ;SET UP STACK POINTER
01800      335
01900      336  001244  000240                      NOP
02000      337  001246  075001                      FADD    R1              ;FLOATING ADD ON THE R1 STACK
02100      338
02200      339  001250  004767  014764              JSR     PC,     POPR    ;POP THE ANSWER
02300      340  001254  010167  177154              MOV     R1,     $SP     ;SAVE ''STACK POINTER''
02400      341  001260  105767  177146              TSTB    $PSW            ;CHECK PS (EXCEPT T BIT)
02500      342  001264  001402                      BEQ     .+6             ;BRANCH IF OK
02600      343  001266  104000                      HLT                     ;PS NOT EQUAL TO 000
02700      344  001270  000013                      13                      ;THE ERROR NUMBER IS 13
02800      345
02900      346  001272  022767  000514  177134    CMP     #STACK4,$SP     ;CHECK THE STACK POINTER (R1)
03000      347  001300  001402                      BEQ     .+6             ;BRANCH IF OK
03100      348  001302  104000                      HLT                     ;STACK POINTER (R1) NOT EQUAL TO #STACK4
03200      349  001304  000014                      14                      ;THE ERROR NUMBER IS 14
03300      350
03400      351  001306  022767  040400  177122    CMP     #040400,ANS1    ;CHECK FIRST HALF OF ANSWER
03500      352  001314  001402                      BEQ     .+6             ;BRANCH IF OK
03600      353  001316  104002                      HLT+2                   ;ANS1 NOT EQUAL TO 040400
03700      354  001320  000015                      15                      ;THE ERROR NUMBER IS 15
03800      355
03900      356  001322  005767  177112              TST     ANS2            ;CHECK SECOND HALF OF ANSWER
04000      357  001326  001402                      BEQ     .+6             ;BRANCH IF OK
04100      358  001330  104002                      HLT+2                   ;ANS2 NOT EQUAL TO 000000
04200      359  001332  000016                      16                      ;THE ERROR NUMBER IS 16
04300      360
04400      361  001334  122767  000003  177042    END3:     CMPB    #3,     $TESTN  ;CHECK THE TEST NUMBER
04500      362  001342  001402                      BEQ     .+6             ;BRANCH IF OK
04600      363  001344  104000                      HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
04700      364  001346  000017                      17                      ;THE ERROR NUMBER IS 17
04800      365
04900      366
```

```
00100
00200
00300
00400
00500
00600
00700
00800
00900
01000
01100
01200
01300
01400
01500
01600
01700
01800
01900
02000
02100
02200
02300
02400
02500
02600
02700
02800
02900
03000
03100
03200
03300
03400
03500
03600
03700
03800
03900
04000
04100
04200
04300
04400
04500
04600
04700
04800
04900
```

```
367
368                     ;*****************************************************************************
369                     ;TEST 4:        FADD (LSI-11 FLOATING ADD INSTRUCTION)
370                     ;       077777,177777 + 177777,177777 = 000000,000000
371                     ;       PS = 004,       STACK POINTER = R2
372                     ;*****************************************************************************
373
374  001350 104400              SCOPE
375  001352 004567 014630 TST4: JSR     R5,     PUSHR   ;PUSH 4 WORDS ONTO R2 STACK, SET PRIORITY
376  001356 177777 177777       .WORD   177777,177777   ;SECOND OPERAND ON TOP
377  001362 077777 177777       .WORD   077777,177777   ;FIRST OPERAND ON BOTTOM
378  001366 000100              .WORD   100             ;PROCESSOR PRIORITY LEVEL
379  001370 016456 000340       .WORD   TRAPER,340      ;FIS TRAP VECTOR
380  001374 012702 000510       MOV     #STACK0,R2      ;SET UP STACK POINTER
381
382  001400 000240              NOP
383  001402 075002              FADD    R2              ;FLOATING ADD ON THE R2 STACK
384
385  001404 004767 014630       JSR     PC,     POPR    ;POP THE ANSWER
386  001410 010267 177020       MOV     R2,     $SP     ;SAVE ''STACK POINTER''
387  001414 122767 000004 177010 CMPB    #004,   $PSW    ;CHECK PS (EXCEPT T BIT)
388  001422 001402              BEQ     .+6             ;BRANCH IF OK
389  001424 104000              HLT                     ;PS NOT EQUAL TO 004
390  001426 000020              20                      ;THE ERROR NUMBER IS 20
391
392  001430 022767 000514 176776 CMP     #STACK4,$SP     ;CHECK THE STACK POINTER (R2)
393  001436 001402              BEQ     .+6             ;BRANCH IF OK
394  001440 104000              HLT                     ;STACK POINTER (R2) NOT EQUAL TO #STACK4
395  001442 000021              21                      ;THE ERROR NUMBER IS 21
396
397  001444 005767 176766       TST     ANS1            ;CHECK FIRST HALF OF ANSWER
398  001450 001402              BEQ     .+6             ;BRANCH IF OK
399  001452 104002              HLT+2                   ;ANS1 NOT EQUAL TO 000000
400  001454 000022              22                      ;THE ERROR NUMBER IS 22
401
402  001456 005767 176756       TST     ANS2            ;CHECK SECOND HALF OF ANSWER
403  001462 001402              BEQ     .+6             ;BRANCH IF OK
404  001464 104002              HLT+2                   ;ANS2 NOT EQUAL TO 000000
405  001466 000023              23                      ;THE ERROR NUMBER IS 23
406
407  001470 122767 000004 176706 END4:  CMPB    #4,     $TESTN  ;CHECK THE TEST NUMBER
408  001476 001402              BEQ     .+6             ;BRANCH IF OK
409  001500 104000              HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
410  001502 000024              24                      ;THE ERROR NUMBER IS 24
411
412
```

```
00300
00400     413
00500     414                        ;*****************************************************************************
00600     415                        ;TEST 5:       FADD (LSI-11 FLOATING ADD INSTRUCTION)
00700     416                        ;       052525,052525 + 152525,052524 = 044600,000000
00800     417                        ;       PS = 200,       STACK POINTER = SP
00900     418                        ;*****************************************************************************
01000     419
01100     420  001504  104400                        SCOPE
01200     421  001506  004567  014322                TST5:   JSR     R5,     PUSHS   ;PUSH 4 WORDS ONTO STACK, SET PRIORI
01300     422  001512  152525  052524                .WORD   152525,052524   ;SECOND OPERAND ON TOP
01400     423  001516  052525  052525                .WORD   052525,052525   ;FIRST OPERAND ON BOTTOM
01500     424  001522  000217                        .WORD   217             ;PROCESSOR PRIORITY LEVEL
01600     425  001524  016456  000340                .WORD   TRAPER,340      ;FIS TRAP VECTOR
01700     426
01800     427  001530  000240                        NOP
01900     428  001532  075006                        FADD    SP              ;FLOATING ADD ON THE STACK
02000     429
02100     430  001534  004767  014334                JSR     PC,     POPS    ;POP THE ANSWER
02200     431  001540  022706  000600                CMP     #BEGIN, SP      ;CHECK THE STACK POINTER
02300     432  001544  001405                        BEQ     TSA5            ;BRANCH IF OK
02400     433  001546  012706  000600                MOV     #BEGIN, SP      ;RESTORE STACK POINTER
02500     434  001552  104000                        HLT                     ;STACK POINTER FOULED UP
02600     435  001554  000025                        25                      ;THE ERROR NUMBER IS 25
02700     436  001556  000421                        BR      END5            ;SKIP REST OF TEST
02800     437
02900     438  001560  122767  000200  176644  TSA5: CMPB    #200,   $PSW    ;CHECK PS (EXCEPT T BIT)
03000     439  001566  001402                        BEQ     .+6             ;BRANCH IF OK
03100     440  001570  104000                        HLT                     ;PS NOT EQUAL TO 200
03200     441  001572  000026                        26                      ;THE ERROR NUMBER IS 26
03300     442
03400     443  001574  022767  044600  176634        CMP     #044600,ANS1    ;CHECK FIRST HALF OF ANSWER
03500     444  001602  001402                        BEQ     .+6             ;BRANCH IF OK
03600     445  001604  104002                        HLT+2                   ;ANS1 NOT EQUAL TO 044600
03700     446  001606  000027                        27                      ;THE ERROR NUMBER IS 27
03800     447
03900     448  001610  005767  176624                TST     ANS2            ;CHECK SECOND HALF OF ANSWER
04000     449  001614  001402                        BEQ     .+6             ;BRANCH IF OK
04100     450  001616  104002                        HLT+2                   ;ANS2 NOT EQUAL TO 000000
04200     451  001620  000030                        30                      ;THE ERROR NUMBER IS 30
04300     452
04400     453  001622  122767  000005  176554  END5: CMPB    #5,     $TESTN  ;CHECK THE TEST NUMBER
04500     454  001630  001402                        BEQ     .+6             ;BRANCH IF OK
04600     455  001632  104000                        HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
04700     456  001634  000031                        31                      ;THE ERROR NUMBER IS 31
04800     457
04900     458
```

```
00100
00200
00300
00400     459                                ;*****************************************************************************
00500     460                                ;TEST 6:        FADD (LSI-11 FLOATING ADD INSTRUCTION)
00600     461                                ;       125200,000000 + 025177,177777 = 117200,000000
00700     462                                ;       PS = 210,       STACK POINTER = SP
00800     463                                ;*****************************************************************************
00900     464
01000     465
01100     466  001636  104400                          SCOPE
01200     467  001640  004567  014170                  TST6:   JSR     R5,     PUSHS   ;PUSH 4 WORDS ONTO STACK, SET PRIORI
01300     468  001644  025177  177777                  .WORD   025177,177777   ;SECOND OPERAND ON TOP
01400     469  001650  125200  000000                  .WORD   125200,000000   ;FIRST OPERAND ON BOTTOM
01500     470  001654  000307                          .WORD   307             ;PROCESSOR PRIORITY LEVEL
01600     471  001656  016456  000340                  .WORD   TRAPER,340      ;FIS TRAP VECTOR
01700     472
01800     473  001662  000240                          NOP
01900     474  001664  075006                          FADD    SP              ;FLOATING ADD ON THE STACK
02000     475
02100     476  001666  004767  014202                  JSR     PC,     POPS    ;POP THE ANSWER
02200     477  001672  022706  000600                  CMP     #BEGIN, SP      ;CHECK THE STACK POINTER
02300     478  001676  001405                          BEQ     TSA6            ;BRANCH IF OK
02400     479  001700  012706  000600                  MOV     #BEGIN, SP      ;RESTORE STACK POINTER
02500     480  001704  104000                          HLT                     ;STACK POINTER FOULED UP
02600     481  001706  000032                          32                      ;THE ERROR NUMBER IS 32
02700     482  001710  000421                          BR      END6            ;SKIP REST OF TEST
02800     483
02900     484  001712  122767  000210  176512  TSA6:   CMPB    #210,   $PSW    ;CHECK PS (EXCEPT T BIT)
03000     485  001720  001402                          BEQ     .+6             ;BRANCH IF OK
03100     486  001722  104000                          HLT                     ;PS NOT EQUAL TO 210
03200     487  001724  000033                          33                      ;THE ERROR NUMBER IS 33
03300     488
03400     489  001726  022767  117200  176502          CMP     #117200,ANS1    ;CHECK FIRST HALF OF ANSWER
03500     490  001734  001402                          BEQ     .+6             ;BRANCH IF OK
03600     491  001736  104002                          HLT+2                   ;ANS1 NOT EQUAL TO 117200
03700     492  001740  000034                          34                      ;THE ERROR NUMBER IS 34
03800     493
03900     494  001742  005767  176472                  TST     ANS2            ;CHECK SECOND HALF OF ANSWER
04000     495  001746  001402                          BEQ     .+6             ;BRANCH IF OK
04100     496  001750  104002                          HLT+2                   ;ANS2 NOT EQUAL TO 000000
04200     497  001752  000035                          35                      ;THE ERROR NUMBER IS 35
04300     498
04400     499  001754  122767  000006  176422  END6:   CMPB    #6,     $TESTN  ;CHECK THE TEST NUMBER
04500     500  001762  001402                          BEQ     .+6             ;BRANCH IF OK
04600     501  001764  104000                          HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
04700     502  001766  000036                          36                      ;THE ERROR NUMBER IS 36
04800     503
04900     504
```

```
00100
00200
00300
00400    505
00500    506                    ;*****************************************************************
00600    507                    ;TEST 7:        FADD (LSI-11 FLOATING ADD INSTRUCTION)
00700    508                    ;       135753,024642 + 100125,052525 = 135753,024642
00800    509                    ;       PS = 210.       STACK POINTER = R5
00900    510                    ;*****************************************************************
01000    511
01100    512 001770 104400                          SCOPE
01200    513 001772 004567 014210           TST7:    JSR     R5,     PUSHR   ;PUSH 4 WORDS ONTO R5 STACK, SET PRIORITY
01300    514 001776 100125 052525                    .WORD   100125,052525   ;SECOND OPERAND ON TOP
01400    515 002002 135753 024642                    .WORD   135753,024642   ;FIRST OPERAND ON BOTTOM
01500    516 002006 000347                           .WORD   347             ;PROCESSOR PRIORITY LEVEL
01600    517 002010 016456 000340                    .WORD   TRAPER,340      ;FIS TRAP VECTOR
01700    518 002014 012705 000510                    MOV     #STACK0,R5      ;SET UP STACK POINTER
01800    519
01900    520 002020 000240                           NOP
02000    521 002022 075005                           FADD    R5              ;FLOATING ADD ON THE R5 STACK
02100    522
02200    523 002024 004767 014210                    JSR     PC,     POPR    ;POP THE ANSWER
02300    524 002030 010567 176400                    MOV     R5,     $SP     ;SAVE "STACK POINTER"
02400    525 002034 122767 000210 176370             CMPB    #210,   $PSW    ;CHECK PS (EXCEPT T BIT)
02500    526 002042 001402                           BEQ     .+6             ;BRANCH IF OK
02600    527 002044 104000                           HLT                     ;PS NOT EQUAL TO 210
02700    528 002046 000037                           37                      ;THE ERROR NUMBER IS 37
02800    529
02900    530 002050 022767 000514 176356             CMP     #STACK4,$SP     ;CHECK THE STACK POINTER (R5)
03000    531 002056 001402                           BEQ     .+6             ;BRANCH IF OK
03100    532 002060 104000                           HLT                     ;STACK POINTER (R5) NOT EQUAL TO #STACK4
03200    533 002062 000040                           40                      ;THE ERROR NUMBER IS 40
03300    534
03400    535 002064 022767 135753 176344             CMP     #135753,ANS1    ;CHECK FIRST HALF OF ANSWER
03500    536 002072 001402                           BEQ     .+6             ;BRANCH IF OK
03600    537 002074 104002                           HLT+2                   ;ANS1 NOT EQUAL TO 135753
03700    538 002076 000041                           41                      ;THE ERROR NUMBER IS 41
03800    539
03900    540 002100 022767 024642 176332             CMP     #024642,ANS2    ;CHECK SECOND HALF OF ANSWER
04000    541 002106 001402                           BEQ     .+6             ;BRANCH IF OK
04100    542 002110 104002                           HLT+2                   ;ANS2 NOT EQUAL TO 024642
04200    543 002112 000042                           42                      ;THE ERROR NUMBER IS 42
04300    544
04400    545 002114 122767 000007 176262    END7:    CMPB    #7,     $TESTN  ;CHECK THE TEST NUMBER
04500    546 002122 001402                           BEQ     .+6             ;BRANCH IF OK
04600    547 002124 104000                           HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
04700    548 002126 000043                           43                      ;THE ERROR NUMBER IS 43
04800    549
04900    550
```

```
00300
00400     551
00500     552
00600     553
00700     554
00800     555
00900     556
01000     557
01100     558  002130  104400                          ;*****************************************************************************
01200     559  002132  004567  014050                  ;TEST 10:       FADD (LSI-11 FLOATING ADD INSTRUCTION)
01300     560  002136  001357  024642                  ;       000052,125252 + 001357,024642 = 001357,024642
01400     561  002142  000052  125252                  ;       PS = 200,       STACK POINTER = R1
01500     562  002146  000257                          ;*****************************************************************************
01600     563  002150  016456  000340
01700     564  002154  012701  000510                  TST10:  SCOPE
01800     565                                                  JSR     R5,     PUSHR   ;PUSH 4 WORDS ONTO R1 STACK, SET PRIORITY
01900     566  002160  000240                                  .WORD   001357,024642   ;SECOND OPERAND ON TOP
02000     567  002162  075001                                  .WORD   000052,125252   ;FIRST OPERAND ON BOTTOM
02100     568                                                  .WORD   257             ;PROCESSOR PRIORITY LEVEL
02200     569  002164  004767  014050                        .WORD   TRAPER,340      ;FIS TRAP VECTOR
02300     570  002170  010167  176240                        MOV     #STACK0,R1      ;SET UP STACK POINTER
02400     571  002174  122767  000200  176230
02500     572  002202  001402                              NOP
02600     573  002204  104000                              FADD    R1              ;FLOATING ADD ON THE R1 STACK
02700     574  002206  000044
02800     575                                              JSR     PC,     POPR    ;POP THE ANSWER
02900     576  002210  022767  000514  176216              MOV     R1,     $SP     ;SAVE ''STACK POINTER''
03000     577  002216  001402                              CMPB    #200,   $PSW    ;CHECK PS (EXCEPT T BIT)
03100     578  002220  104000                              BEQ     .+6             ;BRANCH IF OK
03200     579  002222  000045                              HLT                     ;PS NOT EQUAL TO 200
03300     580                                              44                      ;THE ERROR NUMBER IS 44
03400     581  002224  022767  001357  176204
03500     582  002232  001402                              CMP     #STACK4,$SP     ;CHECK THE STACK POINTER (R1)
03600     583  002234  104002                              BEQ     .+6             ;BRANCH IF OK
03700     584  002236  000046                              HLT                     ;STACK POINTER (R1) NOT EQUAL TO #STACK4
03800     585                                              45                      ;THE ERROR NUMBER IS 45
03900     586  002240  022767  024642  176172
04000     587  002246  001402                              CMP     #001357,ANS1    ;CHECK FIRST HALF OF ANSWER
04100     588  002250  104002                              BEQ     .+6             ;BRANCH IF OK
04200     589  002252  000047                              HLT+2                   ;ANS1 NOT EQUAL TO 001357
04300     590                                              46                      ;THE ERROR NUMBER IS 46
04400     591  002254  122767  000010  176122
04500     592  002262  001402                              CMP     #024642,ANS2    ;CHECK SECOND HALF OF ANSWER
04600     593  002264  104000                              BEQ     .+6             ;BRANCH IF OK
04700     594  002266  000050                              HLT+2                   ;ANS2 NOT EQUAL TO 024642
04800     595                                              47                      ;THE ERROR NUMBER IS 47
04900     596
                                         END10:   CMPB    #10.    $TESTN  ;CHECK THE TEST NUMBER
                                                  BEQ     .+6             ;BRANCH IF OK
                                                  HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
                                                  50                      ;THE ERROR NUMBER IS 50
```

```
00100
00200
00300
00400     597
00500     598                    ;*****************************************************************************
00600     599                    ;TEST 11:      FADD (LSI-11 FLOATING ADD INSTRUCTION)
00700     600                    ;      100400,000000 + 000200,000000 = 100200,000000
00800     601                    ;      PS = 010,      STACK POINTER = R5
00900     602                    ;*****************************************************************************
01000     603
01100     604  002270  104400                        SCOPE
01200     605  002272  004567  013710          TST11: JSR     R5,     PUSHR    ;PUSH 4 WORDS ONTO R5 STACK, SET PRIORITY
01300     606  002276  000200  000000                 .WORD  000200,000000     ;SECOND OPERAND ON TOP
01400     607  002302  100400  000000                 .WORD  100400,000000     ;FIRST OPERAND ON BOTTOM
01500     608  002306  000140                        .WORD  140               ;PROCESSOR PRIORITY LEVEL
01600     609  002310  016456  000340                 .WORD  TRAPER,340        ;FIS TRAP VECTOR
01700     610  002314  012705  000510                 MOV    #STACK0,R5        ;SET UP STACK POINTER
01800     611
01900     612  002320  000240                        NOP
02000     613  002322  075005                        FADD   R5                ;FLOATING ADD ON THE R5 STACK
02100     614
02200     615  002324  004767  013710                 JSR    PC,     POPR     ;POP THE ANSWER
02300     616  002330  010567  176100                 MOV    R5,     $SP      ;SAVE "STACK POINTER"
02400     617  002334  122767  000010  176070        CMPB   #010,   $PSW     ;CHECK PS (EXCEPT T BIT)
02500     618  002342  001402                        BEQ    .+6               ;BRANCH IF OK
02600     619  002344  104000                        HLT                      ;PS NOT EQUAL TO 010
02700     620  002346  000051                        51                       ;THE ERROR NUMBER IS 51
02800     621
02900     622  002350  022767  000514  176056        CMP    #STACK4,$SP       ;CHECK THE STACK POINTER (R5)
03000     623  002356  001402                        BEQ    .+6               ;BRANCH IF OK
03100     624  002360  104000                        HLT                      ;STACK POINTER (R5) NOT EQUAL TO #STACK4
03200     625  002362  000052                        52                       ;THE ERROR NUMBER IS 52
03300     626
03400     627  002364  022767  100200  176044        CMP    #100200,ANS1      ;CHECK FIRST HALF OF ANSWER
03500     628  002372  001402                        BEQ    .+6               ;BRANCH IF OK
03600     629  002374  104002                        HLT+2                    ;ANS1 NOT EQUAL TO 100200
03700     630  002376  000053                        53                       ;THE ERROR NUMBER IS 53
03800     631
03900     632  002400  005767  176034                 TST    ANS2              ;CHECK SECOND HALF OF ANSWER
04000     633  002404  001402                        BEQ    .+6               ;BRANCH IF OK
04100     634  002406  104002                        HLT+2                    ;ANS2 NOT EQUAL TO 000000
04200     635  002410  000054                        54                       ;THE ERROR NUMBER IS 54
04300     636
04400     637  002412  122767  000011  175764 END11: CMPB   #11,    $TESTN   ;CHECK THE TEST NUMBER
04500     638  002420  001402                        BEQ    .+6               ;BRANCH IF OK
04600     639  002422  104000                        HLT                      ;WRONG TEST!  PC MUST HAVE FOULED UP.
04700     640  002424  000055                        55                       ;THE ERROR NUMBER IS 55
04800     641
04900     642
```

```
00100
00200
00300
00400     643
00500     644
00600     645
00700     646
00800     647
00900     648
01000     649
01100     650  002426  104400                        ;*****************************************************************************
01200     651  002430  004567  013552                ;TEST 12:       FADD (LSI-11 FLOATING ADD INSTRUCTION)
01300     652  002434  100252  125252                ;       000425,052525 + 100252,125252 = 000200,000000
01400     653  002440  000425  052525                ;       PS = 200,       STACK POINTER = R4
01500     654  002444  000217                        ;*****************************************************************************
01600     655  002446  016456  000340
01700     656  002452  012704  000510                        SCOPE
01800     657                                       TST12:  JSR     R5,     PUSHR   ;PUSH 4 WORDS ONTO R4 STACK, SET PRIORITY
01900     658  002456  000240                                .WORD   100252,125252   ;SECOND OPERAND ON TOP
02000     659  002460  075004                                .WORD   000425,052525   ;FIRST OPERAND ON BOTTOM
02100     660                                               .WORD   217             ;PROCESSOR PRIORITY LEVEL
02200     661  002462  004767  013552                        .WORD   TRAPER,340      ;FIS TRAP VECTOR
02300     662  002466  010467  175742                        MOV     #STACK0,R4      ;SET UP STACK POINTER
02400     663  002472  122767  000200  175732
02500     664  002500  001402                        NOP
02600     665  002502  104000                        FADD    R4                      ;FLOATING ADD ON THE R4 STACK
02700     666  002504  000056
02800     667                                       JSR     PC,     POPR    ;POP THE ANSWER
02900     668  002506  022767  000514  175720        MOV     R4,     $SP     ;SAVE "STACK POINTER"
03000     669  002514  001402                        CMPB    #200,   $PSW    ;CHECK PS (EXCEPT T BIT)
03100     670  002516  104000                        BEQ     .+6             ;BRANCH IF OK
03200     671  002520  000057                        HLT                     ;PS NOT EQUAL TO 200
03300     672                                       56                      ;THE ERROR NUMBER IS 56
03400     673  002522  022767  000200  175706
03500     674  002530  001402                        CMP     #STACK4,$SP     ;CHECK THE STACK POINTER (R4)
03600     675  002532  104002                        BEQ     .+6             ;BRANCH IF OK
03700     676  002534  000060                        HLT                     ;STACK POINTER (R4) NOT EQUAL TO #STACK4
03800     677                                       57                      ;THE ERROR NUMBER IS 57
03900     678  002536  005767  175676
04000     679  002542  001402                        CMP     #000200,ANS1    ;CHECK FIRST HALF OF ANSWER
04100     680  002544  104002                        BEQ     .+6             ;BRANCH IF OK
04200     681  002546  000061                        HLT+2                   ;ANS1 NOT EQUAL TO 000200
04300     682                                       60                      ;THE ERROR NUMBER IS 60
04400     683  002550  122767  000012  175626
04500     684  002556  001402                        TST     ANS2            ;CHECK SECOND HALF OF ANSWER
04600     685  002560  104000                        BEQ     .+6             ;BRANCH IF OK
04700     686  002562  000062                        HLT+2                   ;ANS2 NOT EQUAL TO 000000
04800     687                                       61                      ;THE ERROR NUMBER IS 61
04900     688
                                          END12:  CMPB    #12,    $TESTN  ;CHECK THE TEST NUMBER
                                                  BEQ     .+6             ;BRANCH IF OK
                                                  HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
                                                  62                      ;THE ERROR NUMBER IS 62
```

```
00300
00400     689
00500     690                    ;*****************************************************************************
00600     691                    ;TEST 13:      FADD (LSI-11 FLOATING ADD INSTRUCTION)
00700     692                    ;       100425,052525 + 000252,125252 = 100200,000000
00800     693                    ;       PS = 210,      STACK POINTER = SP
00900     694                    ;*****************************************************************************
01000     695
01100     696  002564  104400                      SCOPE
01200     697  002566  004567  013242              TST13:  JSR     R5,     PUSHS   ;PUSH 4 WORDS ONTO STACK, SET PRIORI
01300     698  002572  000252  125252              .WORD   000252,125252   ;SECOND OPERAND ON TOP
01400     699  002576  100425  052525              .WORD   100425,052525   ;FIRST OPERAND ON BOTTOM
01500     700  002602  000307                      .WORD   307             ;PROCESSOR PRIORITY LEVEL
01600     701  002604  016456  000340              .WORD   TRAPER,340      ;FIS TRAP VECTOR
01700     702
01800     703  002610  000240                      NOP
01900     704  002612  075006                      FADD    SP              ;FLOATING ADD ON THE STACK
02000     705
02100     706  002614  004767  013254              JSR     PC,     POPS    ;POP THE ANSWER
02200     707  002620  022706  000600              CMP     #BEGIN, SP      ;CHECK THE STACK POINTER
02300     708  002624  001405                      BEQ     TSA13           ;BRANCH IF OK
02400     709  002626  012706  000600              MOV     #BEGIN, SP      ;RESTORE STACK POINTER
02500     710  002632  104000                      HLT                     ;STACK POINTER FOULED UP
02600     711  002634  000063                      63                      ;THE ERROR NUMBER IS 63
02700     712  002636  000421                      BR      END13           ;SKIP REST OF TEST
02800     713
02900     714  002640  122767  000210  175564    TSA13:  CMPB    #210,   $PSW    ;CHECK PS (EXCEPT T BIT)
03000     715  002646  001402                      BEQ     .+6             ;BRANCH IF OK
03100     716  002650  104000                      HLT                     ;PS NOT EQUAL TO 210
03200     717  002652  000064                      64                      ;THE ERROR NUMBER IS 64
03300     718
03400     719  002654  022767  100200  175554            CMP     #100200,ANS1    ;CHECK FIRST HALF OF ANSWER
03500     720  002662  001402                      BEQ     .+6             ;BRANCH IF OK
03600     721  002664  104002                      HLT+2                   ;ANS1 NOT EQUAL TO 100200
03700     722  002666  000065                      65                      ;THE ERROR NUMBER IS 65
03800     723
03900     724  002670  005767  175544            TST     ANS2            ;CHECK SECOND HALF OF ANSWER
04000     725  002674  001402                      BEQ     .+6             ;BRANCH IF OK
04100     726  002676  104002                      HLT+2                   ;ANS2 NOT EQUAL TO 000000
04200     727  002700  000066                      66                      ;THE ERROR NUMBER IS 66
04300     728
04400     729  002702  122767  000013  175474    END13:  CMPB    #13,    $TESTN  ;CHECK THE TEST NUMBER
04500     730  002710  001402                      BEQ     .+6             ;BRANCH IF OK
04600     731  002712  104000                      HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
04700     732  002714  000067                      67                      ;THE ERROR NUMBER IS 67.
04800     733
04900     734
```

```
00300
00400     735
00500     736                    ;*****************************************************************************
00600     737                    ;TEST 14:      FADD (LSI-11 FLOATING ADD INSTRUCTION)
00700     738                    ;       077652,125252 + 077452,125252 = 077777,177777
00800     739                    ;       PS = 200,       STACK POINTER = SP
00900     740                    ;*****************************************************************************
01000     741
01100     742  002716  104400                      SCOPE
01200     743  002720  004567  013110              TST14:  JSR     R5,     PUSHS   ;PUSH 4 WORDS ONTO STACK, SET PRIORI
01300     744  002724  077452  125252              .WORD   077452,125252   ;SECOND OPERAND ON TOP
01400     745  002730  077652  125252              .WORD   077652,125252   ;FIRST OPERAND ON BOTTOM
01500     746  002734  000257                      .WORD   257             ;PROCESSOR PRIORITY LEVEL
01600     747  002736  016456  000340              .WORD   TRAPER,340      ;FIS TRAP VECTOR
01700     748
01800     749  002742  000240                      NOP
01900     750  002744  075006                      FADD    SP              ;FLOATING ADD ON THE STACK
02000     751
02100     752  002746  004767  013122              JSR     PC,     POPS    ;POP THE ANSWER
02200     753  002752  022706  000600              CMP     #BEGIN, SP      ;CHECK THE STACK POINTER
02300     754  002756  001405                      BEQ     TSA14           ;BRANCH IF OK
02400     755  002760  012706  000600              MOV     #BEGIN, SP      ;RESTORE STACK POINTER
02500     756  002764  104000                      HLT                     ;STACK POINTER FOULED UP
02600     757  002766  000070                      70                      ;THE ERROR NUMBER IS 70
02700     758  002770  000422                      BR      END14           ;SKIP REST OF TEST
02800     759
02900     760  002772  122767  000200  175432    TSA14:  CMPB    #200,   $PSW    ;CHECK PS (EXCEPT T BIT)
03000     761  003000  001402                      BEQ     .+6             ;BRANCH IF OK
03100     762  003002  104000                      HLT                     ;PS NOT EQUAL TO 200
03200     763  003004  000071                      71                      ;THE ERROR NUMBER IS 71
03300     764
03400     765  003006  022767  077777  175422              CMP     #077777,ANS1    ;CHECK FIRST HALF OF ANSWER
03500     766  003014  001402                      BEQ     .+6             ;BRANCH IF OK
03600     767  003016  104002                      HLT+2                   ;ANS1 NOT EQUAL TO 077777
03700     768  003020  000072                      72                      ;THE ERROR NUMBER IS 72
03800     769
03900     770  003022  022767  177777  175410              CMP     #177777,ANS2    ;CHECK SECOND HALF OF ANSWER
04000     771  003030  001402                      BEQ     .+6             ;BRANCH IF OK
04100     772  003032  104002                      HLT+2                   ;ANS2 NOT EQUAL TO 177777
04200     773  003034  000073                      73                      ;THE ERROR NUMBER IS 73
04300     774
04400     775  003036  122767  000014  175340    END14:  CMPB    #14.    $TESTN  ;CHECK THE TEST NUMBER
04500     776  003044  001402                      BEQ     .+6             ;BRANCH IF OK
04600     777  003046  104000                      HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
04700     778  003050  000074                      74                      ;THE ERROR NUMBER IS 74
04800     779
04900     780
```

```
00300
00400     781                                   ;*************************************************************************
00500     782                                   ;TEST 15:       FADD (LSI-11 FLOATING ADD INSTRUCTION)
00600     783                                   ;       177452,125252 + 177652,125252 = 177777,177777
00700     784                                   ;       PS = 210,       STACK POINTER = R4
00800     785                                   ;*************************************************************************
00900     786
01000     787                                           SCOPE
01100     788  003052  104400                  TST15:   JSR     R5,     PUSHR   ;PUSH 4 WORDS ONTO R4 STACK, SET PRIORITY
01200     789  003054  004567  013126                   .WORD   177652,125252   ;SECOND OPERAND ON TOP
01300     790  003060  177652  125252                   .WORD   177452,125252   ;FIRST OPERAND ON BOTTOM
01400     791  003064  177452  125252                   .WORD   357             ;PROCESSOR PRIORITY LEVEL
01500     792  003070  000357                          .WORD   TRAPER,340      ;FIS TRAP VECTOR
01600     793  003072  016456  000340                   MOV     #STACK0,R4      ;SET UP STACK POINTER
01700     794  003076  012704  000510
01800     795
01900     796  003102  000240                          NOP
02000     797  003104  075004                          FADD    R4              ;FLOATING ADD ON THE R4 STACK
02100     798
02200     799  003106  004767  013126                   JSR     PC,     POPR    ;POP THE ANSWER
02300     800  003112  010467  175316                   MOV     R4,     $SP     ;SAVE ''STACK POINTER''
02400     801  003116  122767  000210  175306         CMPB    #210,   $PSW    ;CHECK PS (EXCEPT T BIT)
02500     802  003124  001402                          BEQ     .+6             ;BRANCH IF OK
02600     803  003126  104000                          HLT                     ;PS NOT EQUAL TO 210
02700     804  003130  000075                          75                      ;THE ERROR NUMBER IS 75
02800     805
02900     806  003132  022767  000514  175274         CMP     #STACK4,$SP     ;CHECK THE STACK POINTER (R4).
03000     807  003140  001402                          BEQ     .+6             ;BRANCH IF OK
03100     808  003142  104000                          HLT                     ;STACK POINTER (R4) NOT EQUAL TO #STACK4
03200     809  003144  000076                          76                      ;THE ERROR NUMBER IS 76
03300     810
03400     811  003146  022767  177777  175262         CMP     #177777,ANS1    ;CHECK FIRST HALF OF ANSWER
03500     812  003154  001402                          BEQ     .+6             ;BRANCH IF OK
03600     813  003156  104002                          HLT+2                   ;ANS1 NOT EQUAL TO 177777
03700     814  003160  000077                          77                      ;THE ERROR NUMBER IS 77
03800     815
03900     816  003162  022767  177777  175250         CMP     #177777,ANS2    ;CHECK SECOND HALF OF ANSWER
04000     817  003170  001402                          BEQ     .+6             ;BRANCH IF OK
04100     818  003172  104002                          HLT+2                   ;ANS2 NOT EQUAL TO 177777
04200     819  003174  000100                          100                     ;THE ERROR NUMBER IS 100
04300     820
04400     821  003176  122767  000015  175200  END15: CMPB    #15,    $TESTN  ;CHECK THE TEST NUMBER
04500     822  003204  001402                          BEQ     .+6             ;BRANCH IF OK
04600     823  003206  104000                          HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
04700     824  003210  000101                          101                     ;THE ERROR NUMBER IS 101
04800     825
04900     826
```

```
00100
00200
00300
00400     827
00500     828                    ;*****************************************************************************
00600     829                    ;TEST 16:       FADD (LSI-11 FLOATING ADD INSTRUCTION)
00700     830                    ;       004000,105004 + 104000,104000 = 000401,000000
00800     831                    ;       PS = 000,       STACK POINTER = PC
00900     832                    ;*****************************************************************************
01000     833
01100     834  003212  104400                        SCOPE
01200     835  003214  004567  013140      TST16:    JSR     R5,     PUSH7   ;PUSH 4 WORDS ONTO STACK, SET PRIORITY
01300     836  003220  003244                        .WORD   STK16           ;TOP OF STACK
01400     837  003222  104000  104000                .WORD   104000,104000   ;SECOND OPERAND ON TOP
01500     838  003226  004000  105004                .WORD   004000,105004   ;FIRST OPERAND ON BOTTOM
01600     839  003232  000144                        .WORD   144             ;PROCESSOR PRIORITY LEVEL
01700     840  003234  016456  000340                .WORD   TRAPER,340      ;FIS TRAP VECTOR
01800     841
01900     842  003240  000240                        NOP
02000     843  003242  075007                        FADD    PC              ;FLOATING ADD ON FOLLOWING 4 WORDS
02100     844  003244  104000              STK16:    104000                  ;SHOULD CONTAIN 104000
02200     845  003246  104000                        104000                  ;SHOULD CONTAIN 104000
02300     846  003250  004000                        004000                  ;BEFORE FADD, 004000; AFTER, 000401
02400     847  003252  105004                        105004                  ;BEFORE FADD, 105004; AFTER, 000000
02500     848
02600     849  003254  004767  013130              JSR     PC,     POP7    ;POP THE ANSWER
02700     850  003260  105767  175146              TSTB    $PSW            ;CHECK PS (EXCEPT T BIT)
02800     851  003264  001402                      BEQ     .+6             ;BRANCH IF OK
02900     852  003266  104000                      HLT                     ;PS NOT EQUAL TO 000
03000     853  003270  000102                      102                     ;THE ERROR NUMBER IS 102
03100     854
03200     855  003272  022767  104000  175136    CMP     #104000,ANS1    ;CHECK FIRST HALF OF INPUT DATA (STK16)
03300     856  003300  001402                      BEQ     .+6             ;BRANCH IF OK
03400     857  003302  104002                      HLT+2                   ;ANS1 NOT EQUAL TO 104000
03500     858  003304  000103                      103                     ;THE ERROR NUMBER IS 103
03600     859
03700     860  003306  022767  104000  175124    CMP     #104000,ANS2    ;CHECK SECOND HALF OF INPUT DATA (STK16+2)
04000     861  003314  001402                      BEQ     .+6             ;BRANCH IF OK
04100     862  003316  104002                      HLT+2                   ;ANS2 NOT EQUAL TO 104000
04200     863  003320  000104                      104                     ;THE ERROR NUMBER IS 104
04300     864
04400     865  003322  022767  000401  175112    CMP     #000401,ANS3    ;CHECK FIRST HALF OF ANSWER
04500     866  003330  001402                      BEQ     .+6             ;BRANCH IF OK
04600     867  003332  104004                      HLT+4                   ;ANS3 NOT EQUAL TO 000401
04700     868  003334  000105                      105                     ;THE ERROR NUMBER IS 105
05000     869
05100     870  003336  005767  175102              TST     ANS4            ;CHECK SECOND HALF OF ANSWER
05200     871  003342  001402                      BEQ     .+6             ;BRANCH IF OK
05300     872  003344  104004                      HLT+4                   ;ANS4 NOT EQUAL TO 000000
05400     873  003346  000106                      106                     ;THE ERROR NUMBER IS 106
05500     874
05600     875  003350  122767  000016  175026    END16:    CMPB    #16,    $TESTN  ;CHECK THE TEST NUMBER
05700     876  003356  001402                      BEQ     .+6             ;BRANCH IF OK
05800     877  003360  104000                      HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
05900     878  003362  000107                      107                     ;THE ERROR NUMBER IS 107
06000     879
06100     880
06200     881
06300     882                    ;*****************************************************************************
```

```
00100   CVKACC  MACY11 30A(1052)  21-AUG-78  15:28  PAGE 22                                                    SEQ 0030
00200   CVKACC.P11    16-AUG-78 08:41          TEST FLOATING ADD INSTRUCTION WITH UNDERFLOW
00300
00400      883                                          ;TEST 17:       FADD (LSI-11 FLOATING ADD INSTRUCTION)
00500      884                                          ;       100200,000000 + 000377,177777 ==> UNDERFLOW
00600      885                                          ;       PS(ON STACK) = 012,     STACK POINTER = R3
00700      886                                          ;*********************************************************************
00800      887
00900      888  003364  104400                                  SCOPE
01000      889  003366  004567  012614                TST17:    JSR     R5,     PUSHR   ;PUSH 4 WORDS ONTO R3 STACK, SET PRIORITY
01100      890  003372  000377  177777                          .WORD   000377,177777   ;SECOND OPERAND ON TOP
01200      891  003376  100200  000000                          .WORD   100200,000000   ;FIRST OPERAND ON BOTTOM
01300      892  003402  000157                                .WORD   157             ;PROCESSOR PRIORITY LEVEL
01400      893  003404  003436  000000                          .WORD   ISR17,  000     ;FIS TRAP VECTOR
01500      894  003410  012703  000510                          MOV     #STACK0,R3      ;SET UP R3 AS STACK POINTER
01600      895
01700      896  003414  000240                                NOP
01800      897  003416  075003                                FADD    R3              ;FLOATING ADD ON THE R3 STACK
01900      898
02000      899  003420  004767  012614                RTA17:    JSR     PC,     POPR    ;POP THE ''ANSWER''
02100      900  003424  010367  175004                          MOV     R3,     $SP     ;SAVE STACK POINTER (R3)
02200      901  003430  104002                                HLT+2                   ;FIS TRAP DIDN'T OCCURE!
02300      902  003432  000110                                110                     ;THE ERROR NUMBER IS 110
02400      903  003434  000462                                BR      END17
02500      904
02600      905  003436  004767  012626                ISR17:    JSR     PC,     POPER   ;POP ALL DATA OFF THE STACKS
02700      906  003442  010367  174766                          MOV     R3,     $SP     ;SAVE STACK POINTER (R3)
02800      907  003446  105767  174760                          TSTB    $PSW            ;CHECK PS AFTER FIS TRAP
02900      908  003452  001402                                BEQ     .+6             ;BRANCH IF OK
03000      909  003454  104000                                HLT                     ;PS AFTER FIS TRAP NOT EQUAL TO 000
03100      910  003456  000111                                111                     ;THE ERROR NUMBER IS 111
03200      911
03300      912  003460  022767  000510  174746                CMP     #STACK0,$SP     ;CHECK THE STACK POINTER (R3)
03400      913  003466  001402                                BEQ     .+6             ;BRANCH IF OK
03500      914  003470  104000                                HLT                     ;STACK POINTER (R3) NOT EQUAL TO #STACK0
03600      915  003472  000112                                112                     ;THE ERROR NUMBER IS 112
03700      916
03800      917  003474  022767  003420  174734                CMP     #RTA17, ANS1    ;CHECK FIS TRAP RETURN ADDRESS
03900      918  003502  001402                                BEQ     .+6             ;BRANCH IF OK
04000      919  003504  104001                                HLT+1                   ;FIS TRAP AT WRONG ADDRESS
04100      920  003506  000113                                113                     ;THE ERROR NUMBER IS 113
04200      921
04300      922  003510  022767  000012  174722                CMP     #012,   ANS2    ;CHECK PS BEFORE FIS TRAP
04400      923  003516  001402                                BEQ     .+6             ;BRANCH IF OK
04500      924  003520  104002                                HLT+2                   ;PS AT FIS TRAP TIME NOT 012
04600      925  003522  000114                                114                     ;THE ERROR NUMBER IS 114
04700      926
04800      927  003524  022767  000377  174710                CMP     #000377,ANS3    ;CHECK DATA FROM THE STACK
04900      928  003532  001402                                BEQ     .+6             ;BRANCH IF OK
05000      929  003534  104004                                HLT+4                   ;DATA ON STACK (000377) CHANGED
05100      930  003536  000115                                115                     ;THE ERROR NUMBER IS 115
05200      931
05300      932  003540  022767  177777  174676                CMP     #177777,ANS4    ;CHECK DATA FROM STACK
05400      933  003546  001402                                BEQ     .+6             ;BRANCH IF OK
05500      934  003550  104004                                HLT+4                   ;DATA ON STACK (177777) CHANGED
05600      935  003552  000116                                116                     ;THE ERROR NUMBER IS 116
05700      936
05800      937  003554  022767  100200  174664                CMP     #100200,ANS5    ;CHECK DATA FROM STACK
05900      938  003562  001402                                BEQ     .+6             ;BRANCH IF OK
```

```
00100   CVKACC  MACY11 30A(1052)  21-AUG-78  15:28  PAGE 23
00200   CVKACC.P11    16-AUG-78 08:41          TEST FLOATING ADD INSTRUCTION WITH UNDERFLOW                    SEQ 0031
00300
00400       939  003564  104006                                  HLT+6                   ;DATA ON STACK (100200) CHANGED
00500       940  003566  000117                                  117                     ;THE ERROR NUMBER IS 117
00600       941
00700       942  003570  005767  174654                          TST     ANS6            ;CHECK DATA FROM STACK
00800       943  003574  001402                                  BEQ     .+6             ;BRANCH IF OK
00900       944  003576  104006                                  HLT+6                   ;DATA ON STACK (000000) CHANGED
01000       945  003600  000120                                  120                     ;THE ERROR NUMBER IS 120
01100       946
01200       947  003602  122767  000017  174574        END17:  CMPB    #17,    $TESTN  ;CHECK THE TEST NUMBER
01300       948  003610  001402                                  BEQ     .+6             ;BRANCH IF OK
01400       949  003612  104000                                  HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
01500       950  003614  000121                                  121                     ;THE ERROR NUMBER IS 121
01600       951
01700       952
```

```
00100
00200
00300
00400     953
00500     954
00600     955
00700     956
00800     957
00900     958
01000     959
01100     960  003616  104400                          ;*****************************************************************************
01200     961  003620  004567  012210                  ;TEST 20:       FADD (LSI-11 FLOATING ADD INSTRUCTION)
01300     962  003624  100377  177777                  ;       000200,000000 + 100377,177777 ==> UNDERFLOW
01400     963  003630  000200  000000                  ;       PS(ON STACK) = 212,     STACK POINTER = SP
01500     964  003634  000257                          ;*****************************************************************************
01600     965  003636  003664  000340
01700     966                                          SCOPE
01800     967  003642  000240                TST20:    JSR     R5,     PUSHS   ;PUSH 4 WORDS ONTO STACK, SET PRIORITY
01900     968  003644  075006                          .WORD   100377,177777   ;SECOND OPERAND ON TOP
02000     969                                          .WORD   000200,000000   ;FIRST OPERAND ON BOTTOM
02100     970  003646  004767  012222                  .WORD   257             ;PROCESSOR PRIORITY LEVEL
02200     971  003652  104002                          .WORD   ISR20,  340     ;FIS TRAP VECTOR
02300     972  003654  000122
02400     973  003656  012706  000600                  NOP
02500     974  003662  000463                          FADD    SP              ;FLOATING ADD ON THE STACK
02600     975
02700     976  003664  004767  012236        RTA20:    JSR     PC,     POPS    ;POP THE ''ANSWER''
02800     977  003670  022706  000600                  HLT+2                   ;FIS TRAP DIDN'T OCCURE!
02900     978  003674  001405                          122                     ;THE ERROR NUMBER IS 122
03000     979  003676  012706  000600                  MOV     #BEGIN, SP      ;RESTORE THE STACK POINTER
03100     980  003702  104000                          BR      END20
03200     981  003704  000123
03300     982  003706  000451              ISR20:    JSR     PC,     POPES   ;POP ALL DATA OFF THE STACK
03400     983                                          CMP     #BEGIN, SP      ;CHECK THE STACK POINTER
03500     984  003710  122767  000340  174514        BEQ     ISA20           ;BRANCH IF OK
03600     985  003716  001402                          MOV     #BEGIN, SP      ;RESTORE THE STACK POINTER
03700     986  003720  104000                          HLT                     ;STACK POINTER FOULED UP
03800     987  003722  000124                          123                     ;THE ERROR NUMBER IS 123
03900     988                                          BR      END20           ;SKIP REST OF TEST
04000     989  003724  022767  003646  174504
04100     990  003732  001402              ISA20:    CMPB    #340,   $PSW    ;CHECK PS AFTER FIS TRAP
04200     991  003734  104001                          BEQ     .+6             ;BRANCH IF OK
04300     992  003736  000125                          HLT                     ;PS AFTER FIS TRAP NOT EQUAL TO 340
04400     993                                          124                     ;THE ERROR NUMBER IS 124
04500     994  003740  022767  000212  174472
04600     995  003746  001402                          CMP     #RTA20, ANS1    ;CHECK FIS TRAP RETURN ADDRESS
04700     996  003750  104002                          BEQ     .+6             ;BRANCH IF OK
04800     997  003752  000126                          HLT+1                   ;FIS TRAP AT WRONG ADDRESS
04900     998                                          125                     ;THE ERROR NUMBER IS 125
05000     999  003754  022767  100377  174460
05100    1000  003762  001402                          CMP     #212,   ANS2    ;CHECK PS BEFORE FIS TRAP
05200    1001  003764  104004                          BEQ     .+6             ;BRANCH IF OK
05300    1002  003766  000127                          HLT+2                   ;PS AT FIS TRAP TIME NOT 212
05400    1003                                          126                     ;THE ERROR NUMBER IS 126
05500    1004  003770  022767  177777  174446
05600    1005  003776  001402                          CMP     #100377,ANS3    ;CHECK DATA FROM THE STACK
05700    1006  004000  104004                          BEQ     .+6             ;BRANCH IF OK
05800    1007  004002  000130                          HLT+4                   ;DATA ON STACK (100377) CHANGED
05900    1008                                          127                     ;THE ERROR NUMBER IS 127

                                                      CMP     #177777,ANS4    ;CHECK DATA FROM STACK
                                                      BEQ     .+6             ;BRANCH IF OK
                                                      HLT+4                   ;DATA ON STACK (177777) CHANGED
                                                      130                     ;THE ERROR NUMBER IS 130
```

```
00300
00400     1009  004004  022767  000200  174434          CMP     #000200,ANS5    ;CHECK DATA FROM STACK
00500     1010  004012  001402                          BEQ     .+6             ;BRANCH IF OK
00600     1011  004014  104006                          HLT+6                   ;DATA ON STACK (000200) CHANGED
00700     1012  004016  000131                          131                     ;THE ERROR NUMBER IS 131
00800     1013
00900     1014  004020  005767  174424                  TST     ANS6            ;CHECK DATA FROM STACK
01000     1015  004024  001402                          BEQ     .+6             ;BRANCH IF OK
01100     1016  004026  104006                          HLT+6                   ;DATA ON STACK (000000) CHANGED
01200     1017  004030  000132                          132                     ;THE ERROR NUMBER IS 132
01300     1018
01400     1019  004032  122767  000020  174344  END20:  CMPB    #20,    $TESTN  ;CHECK THE TEST NUMBER
01500     1020  004040  001402                          BEQ     .+6             ;BRANCH IF OK
01600     1021  004042  104000                          HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
01700     1022  004044  000133                          133                     ;THE ERROR NUMBER IS 133
01800     1023
01900     1024
```

```
00300
00400    1025
00500    1026                    ;*****************************************************************************
00600    1027                    ;TEST 21:       FADD (LSI-11 FLOATING ADD INSTRUCTION)
00700    1028                    ;       177452,125253 + 177652,125252 ==> OVERFLOW
00800    1029                    ;       PS(ON STACK) = 002,     STACK POINTER = R1
00900    1030                    ;*****************************************************************************
01000    1031
01100    1032  004046  104400              SCOPE
01200    1033  004050  004567  012132  TST21: JSR     R5,     PUSHR   ;PUSH 4 WORDS ONTO R1 STACK, SET PRIORITY
01300    1034  004054  177652  125252          .WORD   177652,125252   ;SECOND OPERAND ON TOP
01400    1035  004060  177452  125253          .WORD   177452,125253   ;FIRST OPERAND ON BOTTOM
01500    1036  004064  000105                  .WORD   105             ;PROCESSOR PRIORITY LEVEL
01600    1037  004066  004120  000252          .WORD   ISR21,  252     ;FIS TRAP VECTOR
01700    1038  004072  012701  000510          MOV     #STACK0,R1      ;SET UP R1 AS STACK POINTER
01800    1039
01900    1040  004076  000240              NOP
02000    1041  004100  075001              FADD    R1                      ;FLOATING ADD ON THE R1 STACK
02100    1042
02200    1043  004102  004767  012132  RTA21: JSR     PC,     POPR    ;POP THE ''ANSWER''
02300    1044  004106  010167  174322          MOV     R1,     $SP     ;SAVE STACK POINTER (R1)
02400    1045  004112  104002                  HLT+2                   ;FIS TRAP DIDN'T OCCURE!
02500    1046  004114  000134                  134                     ;THE ERROR NUMBER IS 134
02600    1047  004116  000464                  BR      END21
02700    1048
02800    1049  004120  004767  012144  ISR21: JSR     PC,     POPER   ;POP ALL DATA OFF THE STACKS
02900    1050  004124  010167  174304          MOV     R1,     $SP     ;SAVE STACK POINTER (R1)
03000    1051  004130  122767  000252  174274        CMPB    #252,   $PSW    ;CHECK PS AFTER FIS TRAP
03100    1052  004136  001402                  BEQ     .+6             ;BRANCH IF OK
03200    1053  004140  104000                  HLT                     ;PS AFTER FIS TRAP NOT EQUAL TO 252
03300    1054  004142  000135                  135                     ;THE ERROR NUMBER IS 135
03400    1055
03500    1056  004144  022767  000510  174262        CMP     #STACK0,$SP     ;CHECK THE STACK POINTER (R1)
03600    1057  004152  001402                  BEQ     .+6             ;BRANCH IF OK
03700    1058  004154  104000                  HLT                     ;STACK POINTER (R1) NOT EQUAL TO #STACK0
03800    1059  004156  000136                  136                     ;THE ERROR NUMBER IS 136
03900    1060
04000    1061  004160  022767  004102  174250        CMP     #RTA21, ANS1    ;CHECK FIS TRAP RETURN ADDRESS
04100    1062  004166  001402                  BEQ     .+6             ;BRANCH IF OK
04200    1063  004170  104001                  HLT+1                   ;FIS TRAP AT WRONG ADDRESS
04300    1064  004172  000137                  137                     ;THE ERROR NUMBER IS 137
04400    1065
04500    1066  004174  022767  000002  174236        CMP     #002,   ANS2    ;CHECK PS BEFORE FIS TRAP
04600    1067  004202  001402                  BEQ     .+6             ;BRANCH IF OK
04700    1068  004204  104002                  HLT+2                   ;PS AT FIS TRAP TIME NOT 002
04800    1069  004206  000140                  #140                    ;THE ERROR NUMBER IS 140
04900    1070
05000    1071  004210  022767  177652  174224        CMP     #177652,ANS3    ;CHECK DATA FROM THE STACK
05100    1072  004216  001402                  BEQ     .+6             ;BRANCH IF OK
05200    1073  004220  104004                  HLT+4                   ;DATA ON STACK (177652) CHANGED
05300    1074  004222  000141                  141                     ;THE ERROR NUMBER IS 141
05400    1075
05500    1076  004224  022767  125252  174212        CMP     #125252,ANS4    ;CHECK DATA FROM STACK
05600    1077  004232  001402                  BEQ     .+6             ;BRANCH IF OK
05700    1078  004234  104004                  HLT+4                   ;DATA ON STACK (125252) CHANGED
05800    1079  004236  000142                  142                     ;THE ERROR NUMBER IS 142
05900    1080
```

```
00100
00200
00300
00400    1081  004240  022767  177452  174200                  CMP     #177452,ANS5    ;CHECK DATA FROM STACK
00500    1082  004246  001402                                  BEQ     .+6             ;BRANCH IF OK
00600    1083  004250  104006                                  HLT+6                   ;DATA ON STACK (177452) CHANGED
00700    1084  004252  000143                                  143                     ;THE ERROR NUMBER IS 143
00800    1085
00900    1086  004254  022767  125253  174166                  CMP     #125253,ANS6    ;CHECK,DATA FROM STACK
01000    1087  004262  001402                                  BEQ     .+6             ;BRANCH IF OK
01100    1088  004264  104006                                  HLT+6                   ;DATA ON STACK (125253) CHANGED
01200    1089  004266  000144                                  144                     ;THE ERROR NUMBER IS 144
01300    1090
01400    1091  004270  122767  000021  174106          END21:  CMPB    #21,    $TESTN  ;CHECK THE TEST NUMBER
01500    1092  004276  001402                                  BEQ     .+6             ;BRANCH IF OK
01600    1093  004300  104000                                  HLT                     ;WRONG TEST!  HC MUST HAVE FOULED UP.
01700    1094  004302  000145                                  145                     ;THE ERROR NUMBER IS 145
01800    1095
01900    1096
```

```
00100
00200
00300
00400    1097
00500    1098                    ;*************************************************************************
00600    1099                    ;TEST 22:      FADD (LSI-11 FLOATING ADD INSTRUCTION)
00700    1100                    ;       077652,125253 + 077452,125252 ==> OVERFLOW
00800    1101                    ;       PS(ON STACK) = 002,     STACK POINTER = SP
00900    1102                    ;*************************************************************************
01000    1103
01100    1104  004304  104400                      SCOPE
01200    1105  004306  004567  011522    TST22:    JSR     R5,     PUSHS   ;PUSH 4 WORDS ONTO STACK, SET PRIORITY
01300    1106  004312  077452  125252              .WORD   077452,125252   ;SECOND OPERAND ON TOP
01400    1107  004316  077652  125253              .WORD   077652,125253   ;FIRST OPERAND ON BOTTOM
01500    1108  004322  000003                      .WORD   003             ;PROCESSOR PRIORITY LEVEL
01600    1109  004324  004352  000344              .WORD   ISR22,  344     ;FIS TRAP VECTOR
01700    1110
01800    1111  004330  000240                      NOP
01900    1112  004332  075006                      FADD    SP              ;FLOATING ADD ON THE STACK
02000    1113
02100    1114  004334  004767  011534    RTA22:    JSR     PC,     POPS    ;POP THE ''ANSWER''
02200    1115  004340  104002                      HLT+2                   ;FIS TRAP DIDN'T OCCURE!
02300    1116  004342  000146                      146                     ;THE ERROR NUMBER IS 146
02400    1117  004344  012706  000600              MOV     #BEGIN, SP      ;RESTORE THE STACK POINTER
02500    1118  004350  000464                      BR      END22
02600    1119
02700    1120  004352  004767  011550    ISR22:    JSR     PC,     POPES   ;POP ALL DATA OFF THE STACK
02800    1121  004356  022706  000600              CMP     #BEGIN, SP      ;CHECK THE STACK POINTER
02900    1122  004362  001405                      BEQ     ISA22           ;BRANCH IF OK
03000    1123  004364  012706  000600              MOV     #BEGIN, SP      ;RESTORE THE STACK POINTER
03100    1124  004370  104000                      HLT                     ;STACK POINTER FOULED UP
03200    1125  004372  000147                      147                     ;THE ERROR NUMBER IS 147
03300    1126  004374  000452                      BR      END22           ;SKIP REST OF TEST
03400    1127
03500    1128  004376  122767  000344  174026    ISA22:    CMPB    #344,   $PSW    ;CHECK PS AFTER FIS TRAP
03600    1129  004404  001402                      BEQ     .+6             ;BRANCH IF OK
03700    1130  004406  104000                      HLT                     ;PS AFTER FIS TRAP NOT EQUAL TO 344
03800    1131  004410  000150                      150                     ;THE ERROR NUMBER IS 150
03900    1132
04000    1133  004412  022767  004334  174016              CMP     #RTA22, ANS1    ;CHECK FIS TRAP RETURN ADDRESS
04100    1134  004420  001402                      BEQ     .+6             ;BRANCH IF OK
04200    1135  004422  104001                      HLT+1                   ;FIS TRAP AT WRONG ADDRESS
04300    1136  004424  000151                      151                     ;THE ERROR NUMBER IS 151
04400    1137
04500    1138  004426  022767  000002  174004              CMP     #002,   ANS2    ;CHECK PS BEFORE FIS TRAP
04600    1139  004434  001402                      BEQ     .+6             ;BRANCH IF OK
04700    1140  004436  104002                      HLT+2                   ;PS AT FIS TRAP TIME NOT 002
04800    1141  004440  000152                      152                     ;THE ERROR NUMBER IS 152
04900    1142
05000    1143  004442  022767  077452  173772              CMP     #077452,ANS3    ;CHECK DATA FROM THE STACK
05100    1144  004450  001402                      BEQ     .+6             ;BRANCH IF OK
05200    1145  004452  104004                      HLT+4                   ;DATA ON STACK (077452) CHANGED
05300    1146  004454  000153                      153                     ;THE ERROR NUMBER IS 153
05400    1147
05500    1148  004456  022767  125252  173760              CMP     #125252,ANS4    ;CHECK DATA FROM STACK
05600    1149  004464  001402                      BEQ     .+6             ;BRANCH IF OK
05700    1150  004466  104004                      HLT+4                   ;DATA ON STACK (125252) CHANGED
05800    1151  004470  000154                      154                     ;THE ERROR NUMBER IS 154
05900    1152
```

```
00100
00200
00300
00400    1153  004472  022767  077652  173746            CMP     #077652,ANS5    ;CHECK DATA FROM STACK
00500    1154  004500  001402                            BEQ     .+6             ;BRANCH IS OK
00600    1155  004502  104006                            HLT+6                   ;DATA ON STACK (077652) CHANGED
00700    1156  004504  000155                            155                     ;THE ERROR NUMBER IS 155
00800    1157
00900    1158  004506  022767  125253  173734            CMP     #125253,ANS6    ;CHECK DATA FROM STACK
01000    1159  004514  001402                            BEQ     .+6             ;BRANCH IF OK
01100    1160  004516  104006                            HLT+6                   ;DATA ON STACK (125253) CHANGED
01200    1161  004520  000156                            156                     ;THE ERROR NUMBER IS 156
01300    1162
01400    1163  004522  122767  000022  173654  END22:    CMPB    #22,    $TESTN  ;CHECK THE TEST NUMBER
01500    1164  004530  001402                            BEQ     .+6             ;BRANCH IF OK
01600    1165  004532  104000                            HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
01700    1166  004534  000157                            157                     ;THE ERROR NUMBER IS 157
01800    1167
01900    1168
```

```
00300
00400    1169
00500    1170
00600    1171
00700    1172
00800    1173
00900    1174
01000    1175
01100    1176  004536  104400                    ;*****************************************************************************
01200    1177  004540  004567  011442            ;TEST 23:       FSUB (LSI-11 FLOATING SUBTRACT INSTRUCTION)
01300    1178  004544  135352  051107            ;       177520,017552 - 135352,051107 = 177520,017552
01400    1179  004550  177520  017552            ;       PS = 010,       STACK POINTER = R1
01500    1180  004554  000040                    ;*****************************************************************************
01600    1181  004556  016456  000340
01700    1182  004562  012701  000510            TST23:  SCOPE
01800    1183                                    TST23:  JSR     R5,     PUSHR   ;PUSH 4 WORDS ONTO R1 STACK, SET PRIORITY
01900    1184  004566  000240                            .WORD   135352,051107   ;SECOND OPERAND ON TOP
02000    1185  004570  075011                            .WORD   177520,017552   ;FIRST OPERAND ON BOTTOM
02100    1186                                            .WORD   040             ;PROCESSOR PRIORITY LEVEL
02200    1187  004572  004767  011442                    .WORD   TRAPER, 340     ;FIS TRAP VECTOR
02300    1188  004576  010167  173632                    MOV     #STACK0,R1      ;SET UP STACK POINTER
02400    1189  004602  122767  000010  173622
02500    1190  004610  001402                            NOP
02600    1191  004612  104000                            FSUB    R1              ;FLOATING SUBTRACT ON THE R1 STACK
02700    1192  004614  000160
02800    1193                                            JSR     PC,     POPR    ;POP THE ANSWER
02900    1194  004616  022767  000514  173610            MOV     R1,     $SP     ;SAVE "STACK POINTER"
03000    1195  004624  001402                            CMPB    #010,   $PSW    ;CHECK PS (EXCEPT T BIT)
03100    1196  004626  104000                            BEQ     .+6             ;BRANCH IF OK
03200    1197  004630  000161                            HLT                     ;PS NOT EQUAL TO 010
03300    1198                                            160                     ;THE ERROR NUMBER IS 160
03400    1199  004632  022767  177520  173576
03500    1200  004640  001402                            CMP     #STACK4,$SP     ;CHECK THE STACK POINTER (R1)
03600    1201  004642  104002                            BEQ     .+6             ;BRANCH IF OK
03700    1202  004644  000162                            HLT                     ;STACK POINTER (R1) NOT EQUAL TO #STACK4
03800    1203                                            161                     ;THE ERROR NUMBER IS 161
03900    1204  004646  022767  017552  173564
04000    1205  004654  001402                            CMP     #177520,ANS1    ;CHECK FIRST HALF OF ANSWER
04100    1206  004656  104002                            BEQ     .+6             ;BRANCH IF OK
04200    1207  004660  000163                            HLT+2                   ;ANS1 NOT EQUAL TO 177520
04300    1208                                            162                     ;THE ERROR NUMBER IS 162
04400    1209  004662  122767  000023  173514
04500    1210  004670  001402                            CMP     #017552,ANS2    ;CHECK SECOND HALF OF ANSWER
04600    1211  004672  104000                            BEQ     .+6             ;BRANCH IF OK
04700    1212  004674  000164                            HLT+2                   ;ANS2 NOT EQUAL TO 017552
04800    1213                                            163                     ;THE ERROR NUMBER IS 163
04900    1214
                                            END23:  CMPB    #23.    $TESTN  ;CHECK THE TEST NUMBER
                                                    BEQ     .+6             ;BRANCH IF OK
                                                    HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
                                                    164                     ;THE ERROR NUMBER IS 164
```

```
00100
00200
00300
00400    1215
00500    1216
00600    1217
00700    1218
00800    1219
00900    1220
01000    1221
01100    1222  004676  104400                    ;*****************************************************************************
01200    1223  004700  004567  011302            ;TEST 24:       FSUB (LSI-11 FLOATING SUBTRACT INSTRUCTION)
01300    1224  004704  125252  125253            ;       125252,125252 - 125252,125253 = 017400,000000
01400    1225  004710  125252  125252            ;       PS = 000,       STACK POINTER = R0
01500    1226  004714  000047                    ;*****************************************************************************
01600    1227  004716  016456  000340
01700    1228  004722  012700  000510                    SCOPE
01800    1229                                    TST24:  JSR     R5,     PUSHR   ;PUSH 4 WORDS ONTO R0 STACK, SET PRIORITY
01900    1230  004726  000240                            .WORD   125252,125253   ;SECOND OPERAND ON TOP
02000    1231  004730  075010                            .WORD   125252,125252   ;FIRST OPERAND ON BOTTOM
02100    1232                                            .WORD   047             ;PROCESSOR PRIORITY LEVEL
02200    1233  004732  004767  011302                    .WORD   TRAPER, 340     ;FIS TRAP VECTOR
02300    1234  004736  010067  173472                    MOV     #STACK0,R0      ;SET UP STACK POINTER
02400    1235  004742  105767  173464
02500    1236  004746  001402                    NOP
02600    1237  004750  104000                    FSUB    R0                      ;FLOATING SUBTRACT ON THE R0 STACK
02700    1238  004752  000165
02800    1239                                    JSR     PC,     POPR    ;POP THE ANSWER
02900    1240  004754  022767  000514  173452    MOV     R0,     $SP     ;SAVE ''STACK POINTER''
03000    1241  004762  001402                    TSTB    $PSW            ;CHECK PS (EXCEPT T BIT)
03100    1242  004764  104000                    BEQ     .+6             ;BRANCH IF OK
03200    1243  004766  000166                    HLT                     ;PS NOT EQUAL TO 000
03300    1244                                    165                     ;THE ERROR NUMBER IS 165
03400    1245  004770  022767  017400  173440    CMP     #STACK4,$SP     ;CHECK THE STACK POINTER (R0)
03500    1246  004776  001402                    BEQ     .+6             ;BRANCH IF OK
03600    1247  005000  104002                    HLT                     ;STACK POINTER (R0) NOT EQUAL TO #STACK4
03700    1248  005002  000167                    166                     ;THE ERROR NUMBER IS 166
03800    1249
03900    1250  005004  005767  173430            CMP     #017400,ANS1    ;CHECK FIRST HALF OF ANSWER
04000    1251  005010  001402                    BEQ     .+6             ;BRANCH IF OK
04100    1252  005012  104002                    HLT+2                   ;ANS1 NOT EQUAL TO 017400
04200    1253  005014  000170                    167                     ;THE ERROR NUMBER IS 167
04300    1254
04400    1255  005016  122767  000024  173360    TST     ANS2            ;CHECK SECOND HALF OF ANSWER
04500    1256  005024  001402                    BEQ     .+6             ;BRANCH IF OK
04600    1257  005026  104000                    HLT+2                   ;ANS2 NOT EQUAL TO 000000
04700    1258  005030  000171                    170                     ;THE ERROR NUMBER IS 170
04800    1259
04900    1260    END24:  CMPB    #24,    $TESTN  ;CHECK THE TEST NUMBER
                         BEQ     .+6             ;BRANCH IF OK
                         HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
                         171                     ;THE ERROR NUMBER IS 171
```

```
00300
00400    1261
00500    1262                    ;*****************************************************************************
00600    1263                    ;TEST 25:       FSUB (LSI-11 FLOATING SUBTRACT INSTRUCTION)
00700    1264                    ;       002460,123456 - 100177,177777 = 002460,123456
00800    1265                    ;       PS = 000,       STACK POINTER = SP
00900    1266                    ;*****************************************************************************
01000    1267
01100    1268  005032  104400                      SCOPE
01200    1269  005034  004567  010774    TST25:    JSR     R5,     PUSHS   ;PUSH 4 WORDS ONTO STACK, SET PRIORITY
01300    1270  005040  100177  177777              .WORD   100177,177777   ;SECOND OPERAND ON TOP
01400    1271  005044  002460  123456              .WORD   002460,123456   ;FIRST OPERAND ON BOTTOM
01500    1272  005050  000015                      .WORD   015             ;PROCESSOR PRIORITY LEVEL
01600    1273  005052  016456  000340              .WORD   TRAPER, 340     ;FIS TRAP VECTOR
01700    1274
01800    1275  005056  000240                      NOP
01900    1276  005060  075016                      FSUB    SP              ;FLOATING SUBTRACT ON THE STACK
02000    1277
02100    1278  005062  004767  011006            JSR     PC,     POPS    ;POP THE ANSWER
02200    1279  005066  022706  000600            CMP     #BEGIN, SP      ;CHECK THE STACK POINTER
02300    1280  005072  001405                    BEQ     TSA25           ;BRANCH IF OK
02400    1281  005074  012706  000600            MOV     #BEGIN, SP      ;RESTORE STACK POINTER
02500    1282
02600    1283
02700    1284
02800    1285
02900    1286
03000    1287
03100    1288  005100  104000                    HLT                     ;STACK POINTER FOULED UP
03200    1289  005102  000172                    172                     ;THE ERROR NUMBER IS 172
03300    1290  005104  000422                    BR      END25           ;SKIP REST OF TEST
03400    1291
03500    1292  005106  122767  000000  173316    TSA25:  CMPB    #000,   $PSW    ;CHECK PS (EXCEPT T BIT)
03600    1293  005114  001402                    BEQ     .+6             ;BRANCH IF OK
03700    1294  005116  104000                    HLT                     ;PS NOT EQUAL TO 000
03800    1295  005120  000173                    173                     ;THE ERROR NUMBER IS 173
03900    1296
04000    1297  005122  022767  002460  173306            CMP     #002460,ANS1    ;CHECK FIRST HALF OF ANSWER
04100    1298  005130  001402                    BEQ     .+6             ;BRANCH IF OK
04200    1299  005132  104002                    HLT+2                   ;ANS1 NOT EQUAL TO 002460
04300    1300  005134  000174                    174                     ;THE ERROR NUMBER IS 174
04400    1301
04500    1302  005136  022767  123456  173274            CMP     #123456,ANS2    ;CHECK SECOND HALF OF ANSWER
04600    1303  005144  001402                    BEQ     .+6             ;BRANCH IF OK
04700    1304  005146  104002                    HLT+2                   ;ANS2 NOT EQUAL TO 123456
04800    1305  005150  000175                    175                     ;THE ERROR NUMBER IS 175
04900    1306
05000    1307  005152  122767  000025  173224    END25:  CMPB    #25,    $TESTN  ;CHECK THE TEST NUMBER
05100    1308  005160  001402                    BEQ     .+6             ;BRANCH IF OK
05200    1309  005162  104000                    HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
05300    1310  005164  000176                    176                     ;THE ERROR NUMBER IS 176
05400    1311
05500    1312
```

```
00300
00400    1313
00500    1314
00600    1315
00700    1316
00800    1317
00900    1318
01000    1319
01100    1320  005166  104400                        ;*****************************************************************************
01200    1321  005170  004567  011012                ;TEST 26:       FSUB (LSI-11 FLOATING SUBTRACT INSTRUCTION)
01300    1322  005174  000252  125252                ;       000425,052525 - 000252,125252 = 000200,000000
01400    1323  005200  000425  052525                ;       PS = 200,       STACK POINTER = R4
01500    1324  005204  000217                        ;*****************************************************************************
01600    1325  005206  016456  000340
01700    1326  005212  012704  000510        TST26:  SCOPE
01800    1327                                        JSR     R5,     PUSHR   ;PUSH 4 WORDS ONTO R4 STACK, SET PRIORITY
01900    1328  005216  000240                        .WORD   000252,125252   ;SECOND OPERAND ON TOP
02000    1329  005220  075014                        .WORD   000425,052525   ;FIRST OPERAND ON BOTTOM
02100    1330                                        .WORD   217             ;PROCESSOR PRIORITY LEVEL
02200    1331  005222  004767  011012                .WORD   TRAPER, 340     ;FIS TRAP VECTOR
02300    1332  005226  010467  173202                MOV     #STACK0,R4      ;SET UP STACK POINTER
02400    1333  005232  122767  000200  173172
02500    1334  005240  001402                        NOP
02600    1335  005242  104000                        FSUB    R4              ;FLOATING SUBTRACT ON THE R4 STACK
02700    1336  005244  000177
02800    1337                                        JSR     PC,     POPR    ;POP THE ANSWER
02900    1338  005246  022767  000514  173160        MOV     R4,     $SP     ;SAVE "STACK POINTER"
03000    1339  005254  001402                        CMPB    #200,   $PSW    ;CHECK PS (EXCEPT T BIT)
03100    1340  005256  104000                        BEQ     .+6             ;BRANCH IF OK
03200    1341  005260  000200                        HLT                     ;PS NOT EQUAL TO 200
03300    1342                                        177                     ;THE ERROR NUMBER IS 177
03400    1343  005262  022767  000200  173146
03500    1344  005270  001402                        CMP     #STACK4,$SP     ;CHECK THE STACK POINTER (R4)
03600    1345  005272  104002                        BEQ     .+6             ;BRANCH IF OK
03700    1346  005274  000201                        HLT                     ;STACK POINTER (R4) NOT EQUAL TO #STACK4
03800    1347                                        200                     ;THE ERROR NUMBER IS 200
03900    1348  005276  005767  173136
04000    1349  005302  001402                        CMP     #000200,ANS1    ;CHECK FIRST HALF OF ANSWER
04100    1350  005304  104002                        BEQ     .+6             ;BRANCH IF OK
04200    1351  005306  000202                        HLT+2                   ;ANS1 NOT EQUAL TO 000200
04300    1352                                        201                     ;THE ERROR NUMBER IS 201
04400    1353  005310  122767  000026  173066
04500    1354  005316  001402                        TST     ANS2            ;CHECK SECOND HALF OF ANSWER
04600    1355  005320  104000                        BEQ     .+6             ;BRANCH IF OK
04700    1356  005322  000203                        HLT+2                   ;ANS2 NOT EQUAL TO 000000
04800    1357                                        202                     ;THE ERROR NUMBER IS 202
04900    1358
                                            END26:  CMPB    #26,    $TESTN  ;CHECK THE TEST NUMBER
                                                    BEQ     .+6             ;BRANCH IF OK
                                                    HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
                                                    203                     ;THE ERROR NUMBER IS 203
```

```
00300
00400    1359
00500    1360                    ;*****************************************************************************
00600    1361                    ;TEST 27:       FSUB (LSI-11 FLOATING SUBTRACT INSTRUCTION)
00700    1362                    ;       177452,125252 - 077652,125252 = 177777,177777
00800    1363                    ;       PS = 210,       STACK POINTER = SP
00900    1364                    ;*****************************************************************************
01000    1365
01100    1366  005324  104400                      SCOPE
01200    1367  005326  004567  010502    TST27:    JSR     R5,     PUSHS   ;PUSH 4 WORDS ONTO STACK, SET PRIORITY
01300    1368  005332  077652  125252              .WORD   077652,125252   ;SECOND OPERAND ON TOP
01400    1369  005336  177452  125252              .WORD   177452,125252   ;FIRST OPERAND ON BOTTOM
01500    1370  005342  000357                      .WORD   357             ;PROCESSOR PRIORITY LEVEL
01600    1371  005344  016456  000340              .WORD   TRAPER, 340     ;FIS TRAP VECTOR
01700    1372
01800    1373  005350  000240                      NOP
01900    1374  005352  075016                      FSUB    SP              ;FLOATING SUBTRACT ON THE STACK
02000    1375
02100    1376  005354  004767  010514            JSR     PC,     POPS    ;POP THE ANSWER
02200    1377  005360  022706  000600            CMP     #BEGIN, SP      ;CHECK THE STACK POINTER
02300    1378  005364  001405                    BEQ     TSA27           ;BRANCH IF OK
02400    1379  005366  012706  000600            MOV     #BEGIN, SP      ;RESTORE STACK POINTER
02500    1380
02600    1381
02700    1382
02800    1383
02900    1384
03000    1385
03100    1386  005372  104000                    HLT                     ;STACK POINTER FOULED UP
03200    1387  005374  000204                    204                     ;THE ERROR NUMBER IS 204
03300    1388  005376  000422                    BR      END27           ;SKIP REST OF TEST
03400    1389
03500    1390  005400  122767  000210  173024    TSA27:  CMPB    #210,   $PSW    ;CHECK PS (EXCEPT T BIT)
03600    1391  005406  001402                    BEQ     .+6             ;BRANCH IF OK
03700    1392  005410  104000                    HLT                     ;PS NOT EQUAL TO 210
03800    1393  005412  000205                    205                     ;THE ERROR NUMBER IS 205
03900    1394
04000    1395  005414  022767  177777  173014            CMP     #177777,ANS1    ;CHECK FIRST HALF OF ANSWER
04100    1396  005422  001402                    BEQ     .+6             ;BRANCH IF OK
04200    1397  005424  104002                    HLT+2                   ;ANS1 NOT EQUAL TO 177777
04300    1398  005426  000206                    206                     ;THE ERROR NUMBER IS 206
04400    1399
04500    1400  005430  022767  177777  173002            CMP     #177777,ANS2    ;CHECK SECOND HALF OF ANSWER
04600    1401  005436  001402                    BEQ     .+6             ;BRANCH IF OK
04700    1402  005440  104002                    HLT+2                   ;ANS2 NOT EQUAL TO 177777
04800    1403  005442  000207                    207                     ;THE ERROR NUMBER IS 207
04900    1404
05000    1405  005444  122767  000027  172732    END27:  CMPB    #27,    $TESTN  ;CHECK THE TEST NUMBER
05100    1406  005452  001402                    BEQ     .+6             ;BRANCH IF OK
05200    1407  005454  104000                    HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
05300    1408  005456  000210                    210                     ;THE ERROR NUMBER IS 210
05400    1409
05500    1410
```

```
00300
00400    1411
00500    1412
00600    1413
00700    1414
00800    1415
00900    1416
01000    1417
01100    1418  005460  104400                        ;*********************************************************************
01200    1419  005462  004567  010520                ;TEST 30:      FSUB (LSI-11 FLOATING SUBTRACT INSTRUCTION)
01300    1420  005466  043125  052525                ;       035152,125252 - 043125,052525 = 143125,052524
01400    1421  005472  035152  125252                ;       PS = 010,       STACK POINTER = R3
01500    1422  005476  000147                        ;*********************************************************************
01600    1423  005500  016456  000340
01700    1424  005504  012703  000510                        SCOPE
01800    1425                                        TST30:  JSR     R5,     PUSHR   ;PUSH 4 WORDS ONTO R3 STACK, SET PRIORITY
01900    1426  005510  000240                                .WORD   043125,052525   ;SECOND OPERAND ON TOP
02000    1427  005512  075013                                .WORD   035152,125252   ;FIRST OPERAND ON BOTTOM
02100    1428                                                .WORD   147             ;PROCESSOR PRIORITY LEVEL
02200    1429  005514  004767  010520                        .WORD   TRAPER, 340     ;FIS TRAP VECTOR
02300    1430  005520  010367  172710                        MOV     #STACK0,R3      ;SET UP STACK POINTER
02400    1431  005524  122767  000010  172700
02500    1432  005532  001402                        NOP
02600    1433  005534  104000                        FSUB    R3              ;FLOATING SUBTRACT ON THE R3 STACK
02700    1434  005536  000211
02800    1435                                        JSR     PC,     POPR    ;POP THE ANSWER
02900    1436  005540  022767  000514  172666        MOV     R3,     $SP     ;SAVE ''STACK POINTER''
03000    1437  005546  001402                        CMPB    #010,   $PSW    ;CHECK PS (EXCEPT T BIT)
03100    1438  005550  104000                        BEQ     .+6             ;BRANCH IF OK
03200    1439  005552  000212                        HLT                     ;PS NOT EQUAL TO 010
03300    1440                                        211                     ;THE ERROR NUMBER IS 211
03400    1441  005554  022767  143125  172654
03500    1442  005562  001402                        CMP     #STACK4,$SP     ;CHECK THE STACK POINTER (R3)
03600    1443  005564  104002                        BEQ     .+6             ;BRANCH IF OK
03700    1444  005566  000213                        HLT                     ;STACK POINTER (R3) NOT EQUAL TO #STACK4
03800    1445                                        212                     ;THE ERROR NUMBER IS 212
03900    1446  005570  022767  052524  172642
04000    1447  005576  001402                        CMP     #143125,ANS1    ;CHECK FIRST HALF OF ANSWER
04100    1448  005600  104002                        BEQ     .+6             ;BRANCH IF OK
04200    1449  005602  000214                        HLT+2                   ;ANS1 NOT EQUAL TO 143125
04300    1450                                        213                     ;THE ERROR NUMBER IS 213
04400    1451  005604  122767  000030  172572
04500    1452  005612  001402                        CMP     #052524,ANS2    ;CHECK SECOND HALF OF ANSWER
04600    1453  005614  104000                        BEQ     .+6             ;BRANCH IF OK
04700    1454  005616  000215                        HLT+2                   ;ANS2 NOT EQUAL TO 052524
04800    1455                                        214                     ;THE ERROR NUMBER IS 214
04900    1456
                                            END30:  CMPB    #30,    $TESTN  ;CHECK THE TEST NUMBER
                                                    BEQ     .+6             ;BRANCH IF OK
                                                    HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
                                                    215                     ;THE ERROR NUMBER IS 215
```

```
00100    CVKACC  MACY11 30A(1052)  21-AUG-78  15:28  PAGE 36                                                    SEQ 0044
00200    CVKACC.P11    16-AUG-78 08:41          FSUB TEST SECTION
00300
00400     1457
00500     1458                                  ;*****************************************************************************
00600     1459                                  ;TEST 31:       FSUB (LSI-11 FLOATING SUBTRACT INSTRUCTION)
00700     1460                                  ;       143325,052525 - 135152,125252 = 143325,052525
00800     1461                                  ;       PS = 210,       STACK POINTER = R0
00900     1462                                  ;*****************************************************************************
01000     1463
01100     1464  005620  104400                          SCOPE
01200     1465  005622  004567  010360          TST31:  JSR     R5,     PUSHR   ;PUSH 4 WORDS ONTO R0 STACK, SET PRIORITY
01300     1466  005626  135152  125252                  .WORD   135152,125252   ;SECOND OPERAND ON TOP
01400     1467  005632  143325  052525                  .WORD   143325,052525   ;FIRST OPERAND ON BOTTOM
01500     1468  005636  000243                          .WORD   243             ;PROCESSOR PRIORITY LEVEL
01600     1469  005640  016456  000340                  .WORD   TRAPER, 340     ;FIS TRAP VECTOR
01700     1470  005644  012700  000510                  MOV     #STACK0,R0      ;SET UP STACK POINTER
01800     1471
01900     1472  005650  000240                          NOP
02000     1473  005652  075010                          FSUB    R0              ;FLOATING SUBTRACT ON THE R0 STACK
02100     1474
02200     1475  005654  004767  010360                  JSR     PC,     POPR    ;POP THE ANSWER
02300     1476  005660  010067  172550                  MOV     R0,     $SP     ;SAVE ''STACK POINTER''
02400     1477  005664  122767  000210  172540        CMPB    #210,   $PSW    ;CHECK PS (EXCEPT T BIT)
02500     1478  005672  001402                          BEQ     .+6             ;BRANCH IF OK
02600     1479  005674  104000                          HLT                     ;PS NOT EQUAL TO 210
02700     1480  005676  000216                          216                     ;THE ERROR NUMBER IS 216
02800     1481
02900     1482  005700  022767  000514  172526        CMP     #STACK4,$SP     ;CHECK THE STACK POINTER (R0)
03000     1483  005706  001402                          BEQ     .+6             ;BRANCH IF OK
03100     1484  005710  104000                          HLT                     ;STACK POINTER (R0) NOT EQUAL TO #STACK4
03200     1485  005712  000217                          217                     ;THE ERROR NUMBER IS 217
03300     1486
03400     1487  005714  022767  143325  172514        CMP     #143325,ANS1    ;CHECK FIRST HALF OF ANSWER
03500     1488  005722  001402                          BEQ     .+6             ;BRANCH IF OK
03600     1489  005724  104002                          HLT+2                   ;ANS1 NOT EQUAL TO 143325
03700     1490  005726  000220                          220                     ;THE ERROR NUMBER IS 220
03800     1491
03900     1492  005730  022767  052525  172502        CMP     #052525,ANS2    ;CHECK SECOND HALF OF ANSWER
04000     1493  005736  001402                          BEQ     .+6             ;BRANCH IF OK
04100     1494  005740  104002                          HLT+2                   ;ANS2 NOT EQUAL TO 052525
04200     1495  005742  000221                          221                     ;THE ERROR NUMBER IS 221
04300     1496
04400     1497  005744  122767  000031  172432  END31:  CMPB    #31,    $TESTN  ;CHECK THE TEST NUMBER
04500     1498  005752  001402                          BEQ     .+6             ;BRANCH IF OK
04600     1499  005754  104000                          HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
04700     1500  005756  000222                          222                     ;THE ERROR NUMBER IS 222
04800     1501
04900     1502
```

```
00100
00200
00300
00400    1503
00500    1504                    ;*****************************************************************************
00600    1505                    ;TEST 32:       FSUB (LSI-11 FLOATING SUBTRACT INSTRUCTION)
00700    1506                    ;       135152,125252 - 143325,052525 = 043325,052525
00800    1507                    ;       PS = 200,       STACK POINTER = R5
00900    1508                    ;*****************************************************************************
01000    1509
01100    1510  005760  104400                      SCOPE
01200    1511  005762  004567  010220      TST32:  JSR     R5,     PUSHR   ;PUSH 4 WORDS ONTO R5 STACK, SET PRIORITY
01300    1512  005766  143325  052525              .WORD   143325,052525   ;SECOND OPERAND ON TOP
01400    1513  005772  135152  125252              .WORD   135152,125252   ;FIRST OPERAND ON BOTTOM
01500    1514  005776  000357                      .WORD   357             ;PROCESSOR PRIORITY LEVEL
01600    1515  006000  016456  000340              .WORD   TRAPER, 340     ;FIS TRAP VECTOR
01700    1516  006004  012705  000510              MOV     #STACK0,R5      ;SET UP STACK POINTER
01800    1517
01900    1518  006010  000240                      NOP
02000    1519  006012  075015                      FSUB    R5              ;FLOATING SUBTRACT ON THE R5 STACK
02100    1520
02200    1521  006014  004767  010220'             JSR     PC,     POPR    ;POP THE ANSWER
02300    1522  006020  010567  172410              MOV     R5,     $SP     ;SAVE ''STACK POINTER''
02400    1523  006024  122767  000200  172400      CMPB    #200,   $PSW    ;CHECK PS (EXCEPT T BIT)
02500    1524  006032  001402                      BEQ     .+6             ;BRANCH IF OK
02600    1525  006034  104000                      HLT                     ;PS NOT EQUAL TO 200
02700    1526  006036  000223                      223                     ;THE ERROR NUMBER IS 223
02800    1527
02900    1528  006040  022767  000514  172366      CMP     #STACK4,$SP     ;CHECK THE STACK POINTER (R5)
03000    1529  006046  001402                      BEQ     .+6             ;BRANCH IF OK
03100    1530  006050  104000                      HLT                     ;STACK POINTER (R5) NOT EQUAL TO #STACK4
03200    1531  006052  000224                      224                     ;THE ERROR NUMBER IS 224
03300    1532
03400    1533  006054  022767  043325  172354      CMP     #043325,ANS1    ;CHECK FIRST HALF OF ANSWER
03500    1534  006062  001402                      BEQ     .+6             ;BRANCH IF OK
03600    1535  006064  104002                      HLT+2                   ;ANS1 NOT EQUAL TO 043325
03700    1536  006066  000225                      225                     ;THE ERROR NUMBER IS 225
03800    1537
03900    1538  006070  022767  052525  172342      CMP     #052525,ANS2    ;CHECK SECOND HALF OF ANSWER
04000    1539  006076  001402                      BEQ     .+6             ;BRANCH IF OK
04100    1540  006100  104002                      HLT+2                   ;ANS2 NOT EQUAL TO 052525
04200    1541  006102  000226                      226                     ;THE ERROR NUMBER IS 226
04300    1542
04400    1543  006104  122767  000032  172272  END32:  CMPB    #32,    $TESTN  ;CHECK THE TEST NUMBER
04500    1544  006112  001402                      BEQ     .+6             ;BRANCH IF OK
04600    1545  006114  104000                      HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
04700    1546  006116  000227                      227                     ;THE ERROR NUMBER IS 227
04800    1547
04900    1548
```

```
00300
00400    1549
00500    1550                    ;*****************************************************************************
00600    1551                    ;TEST 33:       FSUB (LSI-11 FLOATING SUBTRACT INSTRUCTION)
00700    1552                    ;       043125,052525 - 035152,125252 = 043125,052524
00800    1553                    ;       PS = 000,       STACK POINTER = R2
00900    1554                    ;*****************************************************************************
01000    1555
01100    1556 006120 104400              SCOPE
01200    1557 006122 004567 010060 TST33: JSR     R5,     PUSHR   ;PUSH 4 WORDS ONTO R2 STACK, SET PRIORITY
01300    1558 006126 035152 125252        .WORD   035152,125252   ;SECOND OPERAND ON TOP
01400    1559 006132 043125 052525        .WORD   043125,052525   ;FIRST OPERAND ON BOTTOM
01500    1560 006136 000040               .WORD   040             ;PROCESSOR PRIORITY LEVEL
01600    1561 006140 016456 000340        .WORD   TRAPER, 340     ;FIS TRAP VECTOR
01700    1562 006144 012702 000510        MOV     #STACK0,R2      ;SET UP STACK POINTER
01800    1563
01900    1564 006150 000240               NOP
02000    1565 006152 075012               FSUB    R2              ;FLOATING SUBTRACT ON THE R2 STACK
02100    1566
02200    1567 006154 004767 010060        JSR     PC,     POPR    ;POP THE ANSWER
02300    1568 006160 010267 172250        MOV     R2,     $SP     ;SAVE ''STACK POINTER''
02400    1569 006164 105767 172242        TSTB    $PSW            ;CHECK PS (EXCEPT T BIT)
02500    1570 006170 001402               BEQ     .+6             ;BRANCH IF OK
02600    1571 006172 104000               HLT                     ;PS NOT EQUAL TO 000
02700    1572 006174 000230               230                     ;THE ERROR NUMBER IS 230
02800    1573
02900    1574 006176 022767 000514 172230 CMP     #STACK4,$SP     ;CHECK THE STACK POINTER (R2)
03000    1575 006204 001402               BEQ     .+6             ;BRANCH IF OK
03100    1576 006206 104000               HLT                     ;STACK POINTER (R2) NOT EQUAL TO #STACK4
03200    1577 006210 000231               231                     ;THE ERROR NUMBER IS 231
03300    1578
03400    1579 006212 022767 043125 172216 CMP     #043125,ANS1    ;CHECK FIRST HALF OF ANSWER
03500    1580 006220 001402               BEQ     .+6             ;BRANCH IF OK
03600    1581 006222 104002               HLT+2                   ;ANS1 NOT EQUAL TO 043125
03700    1582 006224 000232               232                     ;THE ERROR NUMBER IS 232
03800    1583
03900    1584 006226 022767 052524 172204 CMP     #052524,ANS2    ;CHECK SECOND HALF OF ANSWER
04000    1585 006234 001402               BEQ     .+6             ;BRANCH IF OK
04100    1586 006236 104002               HLT+2                   ;ANS2 NOT EQUAL TO 052524
04200    1587 006240 000233               233                     ;THE ERROR NUMBER IS 233
04300    1588
04400    1589 006242 122767 000033 172134 END33:  CMPB    #33,    $TESTN  ;CHECK THE TEST NUMBER
04500    1590 006250 001402               BEQ     .+6             ;BRANCH IF OK
04600    1591 006252 104000               HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
04700    1592 006254 000234               234                     ;THE ERROR NUMBER IS 234
04800    1593
04900    1594
```

```
00300
00400    1595                                   ;*****************************************************************************
00500    1596                                   ;TEST 34:       FSUB (LSI-11 FLOATING SUBTRACT INSTRUCTION)
00600    1597                                   ;       000000,000000 - 000000,000000 = 000000,000000
00700    1598                                   ;       PS = 204,       STACK POINTER = R0
00800    1599                                   ;*****************************************************************************
00900    1600
01000    1601
01100    1602  006256  104400                          SCOPE
01200    1603  006260  004567  007722          TST34:  JSR     R5,     PUSHR   ;PUSH 4 WORDS ONTO R0 STACK, SET PRIORITY
01300    1604  006264  000000  000000                  .WORD   000000,000000   ;SECOND OPERAND ON TOP
01400    1605  006270  000000  000000                  .WORD   000000,000000   ;FIRST OPERAND ON BOTTOM
01500    1606  006274  000217                          .WORD   217             ;PROCESSOR PRIORITY LEVEL
01600    1607  006276  016456  000340                  .WORD   TRAPER, 340     ;FIS TRAP VECTOR
01700    1608  006302  012700  000510                  MOV     #STACK0,R0      ;SET UP STACK POINTER
01800    1609
01900    1610  006306  000240                          NOP
02000    1611  006310  075010                          FSUB    R0              ;FLOATING SUBTRACT ON THE R0 STACK
02100    1612
02200    1613  006312  004767  007722                  JSR     PC,     POPR    ;POP THE ANSWER
02300    1614  006316  010067  172112                  MOV     R0,     $SP     ;SAVE ''STACK POINTER''
02400    1615  006322  122767  000204  172102        CMPB    #204,   $PSW    ;CHECK PS (EXCEPT T BIT)
02500    1616  006330  001402                          BEQ     .+6             ;BRANCH IF OK
02600    1617  006332  104000                          HLT                     ;PS NOT EQUAL TO 204
02700    1618  006334  000235                          235                     ;THE ERROR NUMBER IS 235
02800    1619
02900    1620  006336  022767  000514  172070        CMP     #STACK4,$SP     ;CHECK THE STACK POINTER (R0)
03000    1621  006344  001402                          BEQ     .+6             ;BRANCH IF OK
03100    1622  006346  104000                          HLT                     ;STACK POINTER (R0) NOT EQUAL TO #STACK4
03200    1623  006350  000236                          236                     ;THE ERROR NUMBER IS 236
03300    1624
03400    1625  006352  005767  172060                  TST     ANS1            ;CHECK FIRST HALF OF ANSWER
03500    1626  006356  001402                          BEQ     .+6             ;BRANCH IF OK
03600    1627  006360  104002                          HLT+2                   ;ANS1 NOT EQUAL TO 000000
03700    1628  006362  000237                          237                     ;THE ERROR NUMBER IS 237
03800    1629
03900    1630  006364  005767  172050                  TST     ANS2            ;CHECK SECOND HALF OF ANSWER
04000    1631  006370  001402                          BEQ     .+6             ;BRANCH IF OK
04100    1632  006372  104002                          HLT+2                   ;ANS2 NOT EQUAL TO 000000
04200    1633  006374  000240                          240                     ;THE ERROR NUMBER IS 240
04300    1634
04400    1635  006376  122767  000034  172000  END34:  CMPB    #34,    $TESTN  ;CHECK THE TEST NUMBER
04500    1636  006404  001402                          BEQ     .+6             ;BRANCH IF OK
04600    1637  006406  104000                          HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
04700    1638  006410  000241                          241                     ;THE ERROR NUMBER IS 241
04800    1639
04900    1640
```

```
00100
00200
00300
00400    1641
00500    1642                    ;**********************************************************************
00600    1643                    ;TEST 35:      FSUB (LSI-11 FLOATING SUBTRACT INSTRUCTION)
00700    1644                    ;       000000,000000 - 177777,177777 = 077777,177777
00800    1645                    ;       PS = 000,       STACK POINTER = R2
00900    1646                    ;**********************************************************************
01000    1647
01100    1648  006412  104400                      SCOPE
01200    1649  006414  004567  007566    TST35:    JSR     R5,     PUSHR   ;PUSH 4 WORDS ONTO R2 STACK, SET PRIORITY
01300    1650  006420  177777  177777              .WORD   177777,177777   ;SECOND OPERAND ON TOP
01400    1651  006424  000000  000000              .WORD   000000,000000   ;FIRST OPERAND ON BOTTOM
01500    1652  006430  000100                      .WORD   100             ;PROCESSOR PRIORITY LEVEL
01600    1653  006432  016456  000340              .WORD   TRAPER, 340     ;FIS TRAP VECTOR
01700    1654  006436  012702  000510              MOV     #STACK0,R2      ;SET UP STACK POINTER
01800    1655
01900    1656  006442  000240                      NOP
02000    1657  006444  075012                      FSUB    R2              ;FLOATING SUBTRACT ON THE R2 STACK
02100    1658
02200    1659  006446  004767  007566              JSR     PC,     POPR    ;POP THE ANSWER
02300    1660  006452  010267  171756              MOV     R2,     $SP     ;SAVE ''STACK POINTER''
02400    1661  006456  105767  171750              TSTB    $PSW            ;CHECK PS (EXCEPT T BIT)
02500    1662  006462  001402                      BEQ     .+6             ;BRANCH IF OK
02600    1663  006464  104000                      HLT                     ;PS NOT EQUAL TO 000
02700    1664  006466  000242                      242                     ;THE ERROR NUMBER IS 242
02800    1665
02900    1666  006470  022767  000514  171736      CMP     #STACK4,$SP     ;CHECK THE STACK POINTER (R2)
03000    1667  006476  001402                      BEQ     .+6             ;BRANCH IF OK
03100    1668  006500  104000                      HLT                     ;STACK POINTER (R2) NOT EQUAL TO #STACK4
03200    1669  006502  000243                      243                     ;THE ERROR NUMBER IS 243
03300    1670
03400    1671  006504  022767  077777  171724      CMP     #077777,ANS1    ;CHECK FIRST HALF OF ANSWER
03500    1672  006512  001402                      BEQ     .+6             ;BRANCH IF OK
03600    1673  006514  104002                      HLT+2                   ;ANS1 NOT EQUAL TO 077777
03700    1674  006516  000244                      244                     ;THE ERROR NUMBER IS 244.
03800    1675
03900    1676  006520  022767  177777  171712      CMP     #177777,ANS2    ;CHECK SECOND HALF OF ANSWER
04000    1677  006526  001402                      BEQ     .+6             ;BRANCH IF OK
04100    1678  006530  104002                      HLT+2                   ;ANS2 NOT EQUAL TO 177777
04200    1679  006532  000245                      245                     ;THE ERROR NUMBER IS 245
04300    1680
04400    1681  006534  122767  000035  171642    END35:    CMPB    #35,    $TESTN  ;CHECK THE TEST NUMBER
04500    1682  006542  001402                      BEQ     .+6             ;BRANCH IF OK
04600    1683  006544  104000                      HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
04700    1684  006546  000246                      246                     ;THE ERROR NUMBER IS 246
04800    1685
04900    1686
```

```
00100
00200
00300
00400    1687
00500    1688                    ;*****************************************************************************
00600    1689                    ;TEST 36:       FSUB (LSI-11 FLOATING SUBTRACT INSTRUCTION)
00700    1690                    ;       000000,000000 - 077777,177777 = 177777,177777
00800    1691                    ;       PS = 210,       STACK POINTER = SP
00900    1692                    ;*****************************************************************************
01000    1693
01100    1694  006550  104400                      SCOPE
01200    1695  006552  004567  007256    TST36:    JSR     R5,     PUSHS   ;PUSH 4 WORDS ONTO STACK, SET PRIORITY
01300    1696  006556  077777  177777              .WORD   077777,177777   ;SECOND OPERAND ON TOP
01400    1697  006562  000000  000000              .WORD   000000,000000   ;FIRST OPERAND ON BOTTOM
01500    1698  006566  000217                      .WORD   217             ;PROCESSOR PRIORITY LEVEL
01600    1699  006570  016456  000340              .WORD   TRAPER, 340     ;FIS TRAP VECTOR
01700    1700
01800    1701  006574  000240                      NOP
01900    1702  006576  075016                      FSUB    SP              ;FLOATING SUBTRACT ON THE STACK
02000    1703
02100    1704  006600  004767  007270              JSR     PC,     POPS    ;POP THE ANSWER
02200    1705  006604  022706  000600              CMP     #BEGIN, SP      ;CHECK THE STACK POINTER
02300    1706  006610  001405                      BEQ     TSA36           ;BRANCH IF OK
02400    1707  006612  012706  000600              MOV     #BEGIN, SP      ;RESTORE STACK POINTER
02500    1708
02600    1709
02700    1710
02800    1711
02900    1712
03000    1713
03100    1714  006616  104000                      HLT                     ;STACK POINTER FOULED UP
03200    1715  006620  000247                      247                     ;THE ERROR NUMBER IS 247
03300    1716  006622  000422                      BR      END36           ;SKIP REST OF TEST
03400    1717
03500    1718  006624  122767  000210  171600    TSA36:    CMPB    #210,   $PSW    ;CHECK PS (EXCEPT T BIT)
03600    1719  006632  001402                      BEQ     .+6             ;BRANCH IF OK
03700    1720  006634  104000                      HLT                     ;PS NOT EQUAL TO 210
03800    1721  006636  000250                      250                     ;THE ERROR NUMBER IS 250
03900    1722
04000    1723  006640  022767  177777  171570              CMP     #177777,ANS1    ;CHECK FIRST HALF OF ANSWER
04100    1724  006646  001402                      BEQ     .+6             ;BRANCH IF OK
04200    1725  006650  104002                      HLT+2                   ;ANS1 NOT EQUAL TO 177777
04300    1726  006652  000251                      251                     ;THE ERROR NUMBER IS 251
04400    1727
04500    1728  006654  022767  177777  171556              CMP     #177777,ANS2    ;CHECK SECOND HALF OF ANSWER
04600    1729  006662  001402                      BEQ     .+6             ;BRANCH IF OK
04700    1730  006664  104002                      HLT+2                   ;ANS2 NOT EQUAL TO 177777
04800    1731  006666  000252                      252                     ;THE ERROR NUMBER IS 252
04900    1732
05000    1733  006670  122767  000036  171506    END36:    CMPB    #36,    $TESTN  ;CHECK THE TEST NUMBER
05100    1734  006676  001402                      BEQ     .+6             ;BRANCH IF OK
05200    1735  006700  104000                      HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
05300    1736  006702  000253                      253                     ;THE ERROR NUMBER IS 253
05400    1737
05500    1738
```

```
00100
00200
00300
00400     1739
00500     1740                    ;*****************************************************************************
00600     1741                    ;TEST 37:       FSUB (LSI-11 FLOATING SUBTRACT INSTRUCTION)
00700     1742                    ;       000425,052525 - 000252,125253 ==> UNDERFLOW
00800     1743                    ;       PS(ON STACK) = 212,     STACK POINTER = SP
00900     1744                    ;*****************************************************************************
01000     1745
01100     1746  006704  104400                      SCOPE
01200     1747  006706  004567  007122    TST37:    JSR     R5,     PUSHS   ;PUSH 4 WORDS ONTO STACK, SET PRIORITY
01300     1748  006712  000252  125253              .WORD   000252,125253   ;SECOND OPERAND ON TOP
01400     1749  006716  000425  052525              .WORD   000425,052525   ;FIRST OPERAND ON BOTTOM
01500     1750  006722  000257                      .WORD   257             ;PROCESSOR PRIORITY LEVEL
01600     1751  006724  006752  000340              .WORD   ISR37,  340     ;FIS TRAP VECTOR
01700     1752
01800     1753  006730  000240                      NOP
01900     1754  006732  075016                      FSUB    SP              ;FLOATING SUBTRACT ON THE STACK
02000     1755
02100     1756  006734  004767  007134    RTA37:    JSR     PC,     POPS    ;POP THE "ANSWER"
02200     1757  006740  104002                      HLT+2                   ;FIS TRAP DIDN'T OCCURE!
02300     1758  006742  000254                      254                     ;THE ERROR NUMBER IS 254
02400     1759  006744  012706  000600              MOV     #BEGIN, SP      ;RESTORE THE STACK POINTER
02500     1760  006750  000464                      BR      END37
02600     1761
02700     1762  006752  004767  007150    ISR37:    JSR     PC,     POPES   ;POP ALL DATA OFF THE STACK
02800     1763  006756  022706  000600              CMP     #BEGIN, SP      ;CHECK THE STACK POINTER
02900     1764  006762  001405                      BEQ     ISA37           ;BRANCH IF OK
03000     1765  006764  012706  000600              MOV     #BEGIN, SP      ;RESTORE THE STACK POINTER
03100     1766  006770  104000                      HLT                     ;STACK POINTER FOULED UP
03200     1767  006772  000255                      255                     ;THE ERROR NUMBER IS 255
03300     1768  006774  000452                      BR      END37           ;SKIP REST OF TEST
03400     1769
03500     1770  006776  122767  000340  171426    ISA37:    CMPB    #340,   $PSW    ;CHECK PS AFTER FIS TRAP
03600     1771  007004  001402                      BEQ     .+6             ;BRANCH IF OK
03700     1772  007006  104000                      HLT                     ;PS AFTER FIS TRAP NOT EQUAL TO 340
03800     1773  007010  000256                      256                     ;THE ERROR NUMBER IS 256
03900     1774
04000     1775  007012  022767  006734  171416              CMP     #RTA37, ANS1    ;CHECK FIS TRAP RETURN ADDRESS
04100     1776  007020  001402                      BEQ     .+6             ;BRANCH IF OK
04200     1777  007022  104001                      HLT+1                   ;FIS TRAP AT WRONG ADDRESS
04300     1778  007024  000257                      257                     ;THE ERROR NUMBER IS 257
04400     1779
04500     1780  007026  022767  000212  171404              CMP     #212,   ANS2    ;CHECK PS BEFORE FIS TRAP
04600     1781  007034  001402                      BEQ     .+6             ;BRANCH IF OK
04700     1782  007036  104002                      HLT+2   .               ;PS AT FIS TRAP TIME NOT 212
04800     1783  007040  000260                      260                     ;THE ERROR NUMBER IS 260
04900     1784
05000     1785  007042  022767  000252  171372              CMP     #000252,ANS3    ;CHECK DATA FROM THE STACK
05100     1786  007050  001402                      BEQ     .+6             ;BRANCH IF OK
05200     1787  007052  104004                      HLT+4                   ;DATA ON STACK (000252) CHANGED
05300     1788  007054  000261                      261                     ;THE ERROR NUMBER IS 261
05400     1789
05500     1790  007056  022767  125253  171360              CMP     #125253,ANS4    ;CHECK DATA FROM STACK
05600     1791  007064  001402                      BEQ     .+6             ;BRANCH IF OK
05700     1792  007066  104004                      HLT+4                   ;DATA ON STACK (125253) CHANGED
05800     1793  007070  000262                      262                     ;THE ERROR NUMBER IS 262
05900     1794
```

```
00100
00200
00300
00400    1795  007072  022767  000425  171346              CMP     #000425,ANS5    ;CHECK DATA FROM STACK
00500    1796  007100  001402                              BEQ     .+6             ;BRANCH IF OK
00600    1797  007102  104006                              HLT+6                   ;DATA ON STACK (000425) CHANGED
00700    1798  007104  000263                              263                     ;THE ERROR NUMBER IS 263
00800    1799
00900    1800  007106  022767  052525  171334              CMP     #052525,ANS6    ;CHECK DATA FROM STACK
01000    1801  007114  001402                              BEQ     .+6             ;BRANCH IF OK
01100    1802  007116  104006                              HLT+6                   ;DATA ON STACK (052525) CHANGED
01200    1803  007120  000264                              264                     ;THE ERROR NUMBER IS 264
01300    1804
01400    1805  007122  122767  000037  171254    END37:    CMPB    #37,    $TESTN  ;CHECK THE TEST NUMBER
01500    1806  007130  001402                              BEQ     .+6             ;BRANCH IF OK
01600    1807  007132  104000                              HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
01700    1808  007134  000265                              265                     ;THE ERROR NUMBER IS 265
01800    1809
01900    1810
```

```
00100
00200
00300
00400    1811
00500    1812                    ;*************************************************************************
00600    1813                    ;TEST 40:       FSUB (LSI-11 FLOATING SUBTRACT INSTRUCTION)
00700    1814                    ;       077652,125253 - 177452,125252 ==> OVERFLOW
00800    1815                    ;       PS(ON STACK) = 002,     STACK POINTER = R3
00900    1816                    ;*************************************************************************
01000    1817
01100    1818 007136 104400              SCOPE
01200    1819 007140 004567 007042 TST40: JSR     R5,     PUSHR   ;PUSH 4 WORDS ONTO R3 STACK, SET PRIORITY
01300    1820 007144 177452 125252        .WORD   177452,125252   ;SECOND OPERAND ON TOP
01400    1821 007150 077652 125253        .WORD   077652,125253   ;FIRST OPERAND ON BOTTOM
01500    1822 007154 000015               .WORD   015             ;PROCESSOR PRIORITY LEVEL
01600    1823 007156 007210 000344        .WORD   ISR40,  344     ;FIS TRAP VECTOR
01700    1824 007162 012703 000510        MOV     #STACK0,R3      ;SET UP R3 AS STACK POINTER
01800    1825
01900    1826 007166 000240               NOP
02000    1827 007170 075013               FSUB    R3              ;FLOATING SUBTRACT ON THE R3 STACK
02100    1828
02200    1829 007172 004767 007042 RTA40: JSR     PC,     POPR    ;POP THE ''ANSWER''
02300    1830 007176 010367 171232        MOV     R3,     $SP     ;SAVE STACK POINTER (R3)
02400    1831 007202 104002               HLT+2                   ;FIS TRAP DIDN'T OCCURE!
02500    1832 007204 000266               266                     ;THE ERROR NUMBER IS 266
02600    1833 007206 000464               BR      END40
02700    1834
02800    1835 007210 004767 007054 ISR40: JSR     PC,     POPER   ;POP ALL DATA OFF THE STACKS
02900    1836 007214 010367 171214        MOV     R3,     $SP     ;SAVE STACK POINTER (R3)
03000    1837 007220 122767 000344 171204 CMPB    #344,   $PSW    ;CHECK PS AFTER FIS TRAP
03100    1838 007226 001402               BEQ     .+6             ;BRANCH IF OK
03200    1839 007230 104000               HLT                     ;PS AFTER FIS TRAP NOT EQUAL TO 344
03300    1840 007232 000267               267                     ;THE ERROR NUMBER IS 267
03400    1841
03500    1842 007234 022767 000510 171172 CMP     #STACK0,$SP     ;CHECK THE STACK POINTER (R3)
03600    1843 007242 001402               BEQ     .+6             ;BRANCH IF OK
03700    1844 007244 104000               HLT                     ;STACK POINTER (R3) NOT EQUAL TO #STACK0
03800    1845 007246 000270               270                     ;THE ERROR NUMBER IS 270
03900    1846
04000    1847 007250 022767 007172 171160 CMP     #RTA40, ANS1    ;CHECK FIS TRAP RETURN ADDRESS
04100    1848 007256 001402               BEQ     .+6             ;BRANCH IF OK
04200    1849 007260 104001               HLT+1                   ;FIS TRAP AT WRONG ADDRESS
04300    1850 007262 000271               271                     ;THE ERROR NUMBER IS 271
04400    1851
04500    1852 007264 022767 000002 171146 CMP     #002,   ANS2    ;CHECK PS BEFORE FIS TRAP
04600    1853 007272 001402               BEQ     .+6             ;BRANCH IF OK
04700    1854 007274 104002               HLT+2                   ;PS AT FIS TRAP TIME NOT 002
04800    1855 007276 000272               272                     ;THE ERROR NUMBER IS 272
04900    1856
05000    1857 007300 022767 177452 171134 CMP     #177452,ANS3    ;CHECK DATA FROM THE STACK
05100    1858 007306 001402               BEQ     .+6             ;BRANCH IF OK
05200    1859 007310 104004               HLT+4                   ;DATA ON STACK (177452) CHANGED
05300    1860 007312 000273               273                     ;THE ERROR NUMBER IS 273
05400    1861
05500    1862 007314 022767 125252 171122 CMP     #125252,ANS4    ;CHECK DATA FROM STACK
05600    1863 007322 001402               BEQ     .+6             ;BRANCH IF OK
05700    1864 007324 104004               HLT+4                   ;DATA ON STACK (125252) CHANGED
05800    1865 007326 000274               274                     ;THE ERROR NUMBER IS 274
05900    1866
```

```
00100
00200
00300
00400    1867  007330  022767  077652  171110                  CMP     #077652,ANS5    ;CHECK DATA FROM STACK
00500    1868  007336  001402                                  BEQ     .+6             ;BRANCH IF OK
00600    1869  007340  104006                                  HLT+6                   ;DATA ON STACK (077652) CHANGED
00700    1870  007342  000275                                  275                     ;THE ERROR NUMBER IS 275
00800    1871
00900    1872  007344  022767  125253  171076                  CMP     #125253,ANS6    ;CHECK DATA FROM STACK
01000    1873  007352  001402                                  BEQ     .+6             ;BRANCH IF OK
01100    1874  007354  104006                                  HLT+6                   ;DATA ON STACK (125253) CHANGED
01200    1875  007356  000276                                  276                     ;THE ERROR NUMBER IS 276
01300    1876
01400    1877  007360  122767  000040  171016          END40:  CMPB    #40.    $TESTN  ;CHECK THE TEST NUMBER
01500    1878  007366  001402                                  BEQ     .+6             ;BRANCH IF OK
01600    1879  007370  104000                                  HLT                     ;WRONG TEST ! PC MUST HAVE FOULED UP.
01700    1880  007372  000277                                  277                     ;THE ERROR NUMBER IS 277
01800    1881
01900    1882
```

```
00100
00200
00300
00400     1883
00500     1884
00600     1885
00700     1886
00800     1887
00900     1888
01000     1889
01100     1890  007374  104400                        ;*********************************************************************
01200     1891  007376  004567  006604                ;TEST 41:       FMUL (LSI-11 FLOATING MULTIPLY INSTRUCTION)
01300     1892  007402  000000  000000                ;       000000,000000 * 000000,000000 = 000000,000000
01400     1893  007406  000000  000000                ;       PS = 004,       STACK POINTER = R4
01500     1894  007412  000111                        ;*********************************************************************
01600     1895  007414  016456  000340
01700     1896  007420  012704  000510                        SCOPE
01800     1897                                        TST41:  JSR     R5,     PUSHR   ;PUSH 4 WORDS ONTO R4 STACK, SET PRIORITY
01900     1898  007424  000240                                .WORD   000000,000000   ;SECOND OPERAND ON TOP
02000     1899  007426  075024                                .WORD   000000,000000   ;FIRST OPERAND ON BOTOM
02100     1900                                                .WORD   111             ;PROCESSOR PRIORITY LEVEL
02200     1901  007430  004767  006604                        .WORD   TRAPER, 340     ;FIS TRAP VECTOR
02300     1902  007434  010467  170774                        MOV     #STACK0,R4      ;SET UP STACK POINTER
02400     1903  007440  122767  000004  170764
02500     1904  007446  001402                        NOP
02600     1905  007450  104000                        FMUL    R4                      ;FLOATING MULTIPLY ON THE R4 STACK
02700     1906  007452  000300
02800     1907                                        JSR     PC,     POPR    ;POP THE ANSWER
02900     1908  007454  022767  000514  170752        MOV     R4,     $SP     ;SAVE ''STACK POINTER''
03000     1909  007462  001402                        CMPB    #004,   $PSW    ;CHECK PS (EXCEPT T BIT)
03100     1910  007464  104000                        BEQ     .+6             ;BRANCH IF OK
03200     1911  007466  000301                        HLT                     ;PS NOT EQUAL TO 004
03300     1912                                        300                     ;THE ERROR NUMBER IS 300
03400     1913  007470  005767  170742
03500     1914  007474  001402                        CMP     #STACK4,$SP     ;CHECK THE STACK POINTER (R4)
03600     1915  007476  104002                        BEQ     .+6             ;BRANCH IF OK
03700     1916  007500  000302                        HLT                     ;STACK POINTER (R4) NOT EQUAL TO #STACK4
03800     1917                                        301                     ;THE ERROR NUMBER IS 301
03900     1918  007502  005767  170732
04000     1919  007506  001402                        TST     ANS1            ;CHECK FIRST HALF OF ANSWER
04100     1920  007510  104002                        BEQ     .+6             ;BRANCH IF OK
04200     1921  007512  000303                        HLT+2                   ;ANS1 NOT EQUAL TO 000000
04300     1922                                        302                     ;THE ERROR NUMBER IS 302
04400     1923  007514  122767  000041  170662
04500     1924  007522  001402                        TST     ANS2            ;CHECK SECOND HALF OF ANSWER
04600     1925  007524  104000                        BEQ     .+6             ;BRANCH IF OK
04700     1926  007526  000304                        HLT+2                   ;ANS2 NOT EQUAL TO 000000
04800     1927                                        303                     ;THE ERROR NUMBER IS 303
04900     1928
                                            END41:  CMPB    #41,    $TESTN  ;CHECK THE TEST NUMBER
                                                    BEQ     .+6             ;BRANCH IF OK
                                                    HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
                                                    304                     ;THE ERROR NUMBER IS 304
```

```
00300
00400    1929
00500    1930                    ;*****************************************************************************
00600    1931                    ;TEST 42:       FMUL (LSI-11 FLOATING MULTIPLY INSTRUCTION)
00700    1932                    ;       140200,000000 * 052345,123456 = 152345,123456
00800    1933                    ;       PS = 210,       STACK POINTER = R2
00900    1934                    ;*****************************************************************************
01000    1935
01100    1936  007530  104400                      SCOPE
01200    1937  007532  004567  006450    TST42:    JSR     R5,     PUSHR   ;PUSH 4 WORDS ONTO R2 STACK, SET PRIORITY
01300    1938  007536  052345  123456              .WORD   052345,123456   ;SECOND OPERAND ON TOP
01400    1939  007542  140200  000000              .WORD   140200,000000   ;FIRST OPERAND ON BOTOM
01500    1940  007546  000343                      .WORD   343             ;PROCESSOR PRIORITY LEVEL
01600    1941  007550  016456  000340              .WORD   TRAPER, 340     ;FIS TRAP VECTOR
01700    1942  007554  012702  000510              MOV     #STACK0,R2      ;SET UP STACK POINTER
01800    1943
01900    1944  007560  000240                      NOP
02000    1945  007562  075022                      FMUL    R2              ;FLOATING MULTIPLY ON THE R2 STACK
02100    1946
02200    1947  007564  004767  006450              JSR     PC,     POPR    ;POP THE ANSWER
02300    1948  007570  010267  170640              MOV     R2,     $SP     ;SAVE ''STACK POINTER''
02400    1949  007574  122767  000210  170630      CMPB    #210,   $PSW    ;CHECK PS (EXCEPT T BIT)
02500    1950  007602  001402                      BEQ     .+6             ;BRANCH IF OK
02600    1951  007604  104000                      HLT                     ;PS NOT EQUAL TO 210
02700    1952  007606  000305                      305                     ;THE ERROR NUMBER IS 305
02800    1953
02900    1954  007610  022767  000514  170616      CMP     #STACK4,$SP     ;CHECK THE STACK POINTER (R2)
03000    1955  007616  001402                      BEQ     .+6             ;BRANCH IF OK
03100    1956  007620  104000                      HLT                     ;STACK POINTER (R2) NOT EQUAL TO #STACK4
03200    1957  007622  000306                      306                     ;THE ERROR NUMBER IS 306
03300    1958
03400    1959  007624  022767  152345  170604      CMP     #152345,ANS1    ;CHECK FIRST HALF OF ANSWER
03500    1960  007632  001402                      BEQ     .+6             ;BRANCH IF OK
03600    1961  007634  104002                      HLT+2                   ;ANS1 NOT EQUAL TO 152345
03700    1962  007636  000307                      307                     ;THE ERROR NUMBER IS 307
03800    1963
03900    1964  007640  022767  123456  170572      CMP     #123456,ANS2    ;CHECK SECOND HALF OF ANSWER
04000    1965  007646  001402                      BEQ     .+6             ;BRANCH IF OK
04100    1966  007650  104002                      HLT+2                   ;ANS2 NOT EQUAL TO 123456
04200    1967  007652  000310                      310                     ;THE ERROR NUMBER IS 310
04300    1968
04400    1969  007654  122767  000042  170522    END42:    CMPB    #42,    $TESTN  ;CHECK THE TEST NUMBER
04500    1970  007662  001402                      BEQ     .+6             ;BRANCH IF OK
04600    1971  007664  104000                      HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
04700    1972  007666  000311                      311                     ;THE ERROR NUMBER IS 311
04800    1973
04900    1974
```

```
00100    CVKACC  MACY11 30A(1052)  21-AUG-78  15:28  PAGE 48                                                    SEQ 0056
00200    CVKACC.P11    16-AUG-78 08:41          FMUL TEST SECTION
00300
00400     1975
00500     1976                                  ;*********************************************************************
00600     1977                                  ;TEST 43:      FMUL (LSI-11 FLOATING MULTIPLY INSTRUCTION)
00700     1978                                  ;       100125,052525 * 135753,024642 = 000000,000000
00800     1979                                  ;       PS = 004,       STACK POINTER = R5
00900     1980                                  ;*********************************************************************
01000     1981
01100     1982  007670  104400                          SCOPE
01200     1983  007672  004567  006310          TST43:  JSR     R5,     PUSHR   ;PUSH 4 WORDS ONTO R5 STACK, SET PRIORITY
01300     1984  007676  135753  024642                  .WORD   135753,024642   ;SECOND OPERAND ON TOP
01400     1985  007702  100125  052525                  .WORD   100125,052525   ;FIRST OPERAND ON BOTOM
01500     1986  007706  000117                          .WORD   117             ;PROCESSOR PRIORITY LEVEL
01600     1987  007710  016456  000340                  .WORD   TRAPER, 340     ;FIS TRAP VECTOR
01700     1988  007714  012705  000510                  MOV     #STACK0,R5      ;SET UP STACK POINTER
01800     1989
01900     1990  007720  000240                          NOP
02000     1991  007722  075025                          FMUL    R5              ;FLOATING MULTIPLY ON THE R5 STACK
02100     1992
02200     1993  007724  004767  006310                  JSR     PC,     POPR    ;POP THE ANSWER
02300     1994  007730  010567  170500                  MOV     R5,     $SP     ;SAVE ''STACK POINTER''
02400     1995  007734  122767  000004  170470         CMPB    #004,   $PSW    ;CHECK PS (EXCEPT T BIT) .
02500     1996  007742  001402                          BEQ     .+6             ;BRANCH IF OK
02600     1997  007744  104000                          HLT                     ;PS NOT EQUAL TO 004
02700     1998  007746  000312                          312                     ;THE ERROR NUMBER IS 312
02800     1999
02900     2000  007750  022767  000514  170456         CMP     #STACK4,$SP     ;CHECK THE STACK POINTER (R5)
03000     2001  007756  001402                          BEQ     .+6             ;BRANCH IF OK
03100     2002  007760  104000                          HLT                     ;STACK POINTER (R5) NOT EQUAL TO #STACK4
03200     2003  007762  000313                          313                     ;THE ERROR NUMBER IS 313
03300     2004
03400     2005  007764  005767  170446                  TST     ANS1            ;CHECK FIRST HALF OF ANSWER
03500     2006  007770  001402                          BEQ     .+6             ;BRANCH IF OK
03600     2007  007772  104002                          HLT+2                   ;ANS1 NOT EQUAL TO 000000
03700     2008  007774  000314                          314                     ;THE ERROR NUMBER IS 314
03800     2009
03900     2010  007776  005767  170436                  TST     ANS2            ;CHECK SECOND HALF OF ANSWER
04000     2011  010002  001402                          BEQ     .+6             ;BRANCH IF OK
04100     2012  010004  104002                          HLT+2                   ;ANS2 NOT EQUAL TO 000000
04200     2013  010006  000315                          315                     ;THE ERROR NUMBER IS 315
04300     2014
04400     2015  010010  122767  000043  170366  END43:  CMPB    #43,    $TESTN  ;CHECK THE TEST NUMBER
04500     2016  010016  001402                          BEQ     .+6             ;BRANCH IF OK
04600     2017  010020  104000                          HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
04700     2018  010022  000316                          316                     ;THE ERROR NUMBER IS 316
04800     2019
04900     2020
```

```
00100
00200
00300
00400    2021
00500    2022                    ;*****************************************************************************
00600    2023                    ;TEST 44:       FMUL (LSI-11 FLOATING MULTIPLY INSTRUCTION)
00700    2024                    ;       161616,161616 * 000052,125252 = 000000,000000
00800    2025                    ;       PS = 204,       STACK POINTER = R3
00900    2026                    ;*****************************************************************************
01000    2027
01100    2028  010024  104400                      SCOPE
01200    2029  010026  004567  006154    TST44:    JSR     R5,     PUSHR   ;PUSH 4 WORDS ONTO R3 STACK, SET PRIORITY
01300    2030  010032  000052  125252              .WORD   000052,125252   ;SECOND OPERAND ON TOP
01400    2031  010036  161616  161616              .WORD   161616,161616   ;FIRST OPERAND ON BOTOM
01500    2032  010042  000217                      .WORD   217             ;PROCESSOR PRIORITY LEVEL
01600    2033  010044  016456  000340              .WORD   TRAPER, 340     ;FIS TRAP VECTOR
01700    2034  010050  012703  000510              MOV     #STACK0,R3      ;SET UP STACK POINTER
01800    2035
01900    2036  010054  000240                      NOP
02000    2037  010056  075023                      FMUL    R3              ;FLOATING MULTIPLY ON THE R3 STACK
02100    2038
02200    2039  010060  004767  006154              JSR     PC,     POPR    ;POP THE ANSWER
02300    2040  010064  010367  170344              MOV     R3,     $SP     ;SAVE ''STACK POINTER''
02400    2041  010070  122767  000204  170334      CMPB    #204,   $PSW    ;CHECK PS (EXCEPT T BIT)
02500    2042  010076  001402                      BEQ     .+6             ;BRANCH IF OK
02600    2043  010100  104000                      HLT                     ;PS NOT EQUAL TO 204
02700    2044  010102  000317                      317                     ;THE ERROR NUMBER IS 317
02800    2045
02900    2046  010104  022767  000514  170322      CMP     #STACK4,$SP     ;CHECK THE STACK POINTER (R3)
03000    2047  010112  001402                      BEQ     .+6             ;BRANCH IF OK
03100    2048  010114  104000                      HLT                     ;STACK POINTER (R3) NOT EQUAL TO #STACK4
03200    2049  010116  000320                      320                     ;THE ERROR NUMBER IS 320
03300    2050
03400    2051  010120  005767  170312              TST     ANS1            ;CHECK FIRST HALF OF ANSWER
03500    2052  010124  001402                      BEQ     .+6             ;BRANCH IF OK
03600    2053  010126  104002                      HLT+2                   ;ANS1 NOT EQUAL TO 000000
03700    2054  010130  000321                      321                     ;THE ERROR NUMBER IS 321
03800    2055
03900    2056  010132  005767  170302              TST     ANS2            ;CHECK SECOND HALF OF ANSWER
04000    2057  010136  001402                      BEQ     .+6             ;BRANCH IF OK
04100    2058  010140  104002                      HLT+2                   ;ANS2 NOT EQUAL TO 000000
04200    2059  010142  000322                      322                     ;THE ERROR NUMBER IS 322
04300    2060
04400    2061  010144  122767  000044  170232    END44:    CMPB    #44,    $TESTN  ;CHECK THE TEST NUMBER
04500    2062  010152  001402                      BEQ     .+6             ;BRANCH IF OK
04600    2063  010154  104000                      HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
04700    2064  010156  000323                      323                     ;THE ERROR NUMBER IS 323
04800    2065
04900    2066
```

```
00100
00200
00300
00400    2067
00500    2068
00600    2069                    ;*********************************************************************
00700    2070                    ;TEST 45:      FMUL (LSI-11 FLOATING MULTIPLY INSTRUCTION)
00800    2071                    ;       176452,125252 * 041500,000000 = 177777,177777
00900    2072                    ;       PS = 210,       STACK POINTER = SP
01000    2073                    ;*********************************************************************
01100    2074  010160  104400                      SCOPE
01200    2075  010162  004567  005646    TST45:    JSR     R5,     PUSHS   ;PUSH 4 WORDS ONTO STACK, SET PRIORITY
01300    2076  010166  041500  000000              .WORD   041500,000000   ;SECOND OPERAND ON TOP
01400    2077  010172  176452  125252              .WORD   176452,125252   ;FIRST OPERAND ON BOTTOM
01500    2078  010176  000357                      .WORD   357             ;PROCESSOR PRIORITY LEVEL
01600    2079  010200  016456  000340              .WORD   TRAPER, 340     ;FIS TRAP VECTOR
01700    2080
01800    2081  010204  000240                      NOP
01900    2082  010206  075026                      FMUL    SP              ;FLOATING MULTIPLY ON THE STACK
02000    2083
02100    2084  010210  004767  005660              JSR     PC,     POPS    ;POP THE ANSWER
02200    2085  010214  022706  000600              CMP     #BEGIN, SP      ;CHECK THE STACK POINTER
02300    2086  010220  001405                      BEQ     TSA45           ;BRANCH IF OK
02400    2087  010222  012706  000600              MOV     #BEGIN, SP      ;RESTORE STACK POINTER
02500    2088  010226  104000                      HLT                     ;STACK POINTER FOULED UP
02600    2089  010230  000324                      324                     ;THE ERROR NUMBER IS 324
02700    2090  010232  000422                      BR      END45           ;SKIP REST OF TEST
02800    2091
02900    2092  010234  122767  000210  170170    TSA45:    CMPB    #210,   $PSW    ;CHECK PS (EXCEPT T BIT)
03000    2093  010242  001402                      BEQ     .+6             ;BRANCH IF OK
03100    2094  010244  104000                      HLT                     ;PS NOT EQUAL TO 210
03200    2095  010246  000325                      325                     ;THE ERROR NUMBER IS 325
03300    2096
03400    2097  010250  022767  177777  170160              CMP     #177777,ANS1    ;CHECK FIRST HALF OF ANSWER
03500    2098  010256  001402                      BEQ     .+6             ;BRANCH IF OK
03600    2099  010260  104002                      HLT+2                   ;ANS1 NOT EQUAL TO 177777
03700    2100  010262  000326                      326                     ;THE ERROR NUMBER IS 326
03800    2101
03900    2102  010264  022767  177777  170146              CMP     #177777,ANS2    ;CHECK SECOND HALF OF ANSWER
04000    2103  010272  001402                      BEQ     .+6             ;BRANCH IF OK
04100    2104  010274  104002                      HLT+2                   ;ANS2 NOT EQUAL TO 177777
04200    2105  010276  000327                      327                     ;THE ERROR NUMBER IS 327
04300    2106
04400    2107  010300  122767  000045  170076    END45:    CMPB    #45,    $TESTN  ;CHECK THE TEST NUMBER
04500    2108  010306  001402                      BEQ     .+6             ;BRANCH IF OK
04600    2109  010310  104000                      HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
04700    2110  010312  000330                      330                     ;THE ERROR NUMBER IS 330
04800    2111
04900    2112
```

```
00300
00400    2113
00500    2114                    ;*****************************************************************************
00600    2115                    ;TEST 46:       FMUL (LSI-11 FLOATING MULTIPLY INSTRUCTION)
00700    2116                    ;       124252,125252 * 114100,000001 = 000200,000000
00800    2117                    ;       PS = 200,       STACK POINTER = R1
00900    2118                    ;*****************************************************************************
01000    2119
01100    2120  010314  104400                      SCOPE
01200    2121  010316  004567  005664    TST46:    JSR     R5,     PUSHR   ;PUSH 4 WORDS ONTO R1 STACK, SET PRIORITY
01300    2122  010322  114100  000001              .WORD   114100,000001   ;SECOND OPERAND ON TOP
01400    2123  010326  124252  125252              .WORD   124252,125252   ;FIRST OPERAND ON BOTOM
01500    2124  010332  000200                      .WORD   200             ;PROCESSOR PRIORITY LEVEL
01600    2125  010334  016456  000340              .WORD   TRAPER, 340     ;FIS TRAP VECTOR
01700    2126  010340  012701  000510              MOV     #STACK0,R1      ;SET UP STACK POINTER
01800    2127
01900    2128  010344  000240                      NOP
02000    2129  010346  075021                      FMUL    R1              ;FLOATING MULTIPLY ON THE R1 STACK
02100    2130
02200    2131  010350  004767  005664              JSR     PC,     POPR    ;POP THE ANSWER
02300    2132  010354  010167  170054              MOV     R1,     $SP     ;SAVE ''STACK POINTER''
02400    2133  010360  122767  000200  170044      CMPB    #200,   $PSW    ;CHECK PS (EXCEPT T BIT)
02500    2134  010366  001402                      BEQ     .+6             ;BRANCH IF OK
02600    2135  010370  104000                      HLT                     ;PS NOT EQUAL TO 200
02700    2136  010372  000331                      331                     ;THE ERROR NUMBER IS 331
02800    2137
02900    2138  010374  022767  000514  170032      CMP     #STACK4,$SP     ;CHECK THE STACK POINTER (R1)
03000    2139  010402  001402                      BEQ     .+6             ;BRANCH IF OK
03100    2140  010404  104000                      HLT                     ;STACK POINTER (R1) NOT EQUAL TO #STACK4
03200    2141  010406  000332                      332                     ;THE ERROR NUMBER IS 332
03300    2142
03400    2143  010410  022767  000200  170020      CMP     #000200,ANS1    ;CHECK FIRST HALF OF ANSWER
03500    2144  010416  001402                      BEQ     .+6             ;BRANCH IF OK
03600    2145  010420  104002                      HLT+2                   ;ANS1 NOT EQUAL TO 000200
03700    2146  010422  000333                      333                     ;THE ERROR NUMBER IS 333
03800    2147
03900    2148  010424  005767  170010              TST     ANS2            ;CHECK SECOND HALF OF ANSWER
04000    2149  010430  001402                      BEQ     .+6             ;BRANCH IF OK
04100    2150  010432  104002                      HLT+2                   ;ANS2 NOT EQUAL TO 000000
04200    2151  010434  000334                      334                     ;THE ERROR NUMBER IS 334
04300    2152
04400    2153  010436  122767  000046  167740    END46:    CMPB    #46,    $TESTN  ;CHECK THE TEST NUMBER
04500    2154  010444  001402                      BEQ     .+6             ;BRANCH IF OK
04600    2155  010446  104000                      HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
04700    2156  010450  000335                      335                     ;THE ERROR NUMBER IS 335
04800    2157
04900    2158
```

```
00100
00200
00300
00400    2159
00500    2160
00600    2161
00700    2162
00800    2163
00900    2164
01000    2165
01100    2166  010452  104400                    ;*********************************************************************
01200    2167  010454  004567  005700            ;TEST 47:      FSUB (LSI-11 FLOATING SUBTRACT INSTRUCTION)
01300    2168  010460  010504                    ;       104000,105004 - 104000,104000 = 100401,000000
01400    2169  010462  104000  104000            ;       PS = 210.       STACK POINTER = PC
01500    2170  010466  104000  105004            ;*********************************************************************
01600    2171  010472  000252
01700    2172  010474  016456  000340                    SCOPE
01800    2173                              TST47:  JSR     R5,     PUSH7   ;PUSH 4 WORDS ONTO STACK, SET PRIORITY
01900    2174  010500  000240                      .WORD   STK47           ;TOP OF STACK
02000    2175  010502  075017                      .WORD   104000,104000   ;SECOND OPERAND ON TOP
02100    2176  010504  104000              STK47:  .WORD   104000,105004   ;FIRST OPERAND ON BOTTOM
02200    2177  010506  104000                      .WORD   252             ;PROCESSOR PRIORITY LEVEL
02300    2178  010510  104000                      .WORD   TRAPER,340      ;FIS TRAP VECTOR
02400    2179  010512  105004
02500    2180                                    NOP
02600    2181  010514  004767  005670              FSUB    PC              ;FLOATING SUBTRACT ON FOLLOWING 4 WORDS
02700    2182  010520  122767  000210  167704      CMPB    #210,   $PSW    ;CHECK PS (EXCEPT T BIT)
02800    2183  010526  001402                      BEQ     .+6             ;BRANCH IF OK
02900    2184  010530  104000                      HLT                     ;PS NOT EQUAL TO 210
03000    2185  010532  000336                      336                     ;THE ERROR NUMBER IS 336
03100    2186
03200    2187  010534  022767  104000  167674      CMP     #104000,ANS1    ;CHECK FIRST HALF OF INPUT DATA (STK47)
03300    2188  010542  001402                      BEQ     .+6             ;BRANCH IF OK
03400    2189  010544  104002                      HLT+2                   ;ANS1 NOT EQUAL TO 104000
03500    2190  010546  000337                      337                     ;THE ERROR NUMBER IS 337
03600    2191
03700    2192  010550  022767  104000  167662      CMP     #104000,ANS2    ;CHECK SECOND HALF OF INPUT DATA (STK47+2)
03800    2193  010556  001402                      BEQ     .+6             ;BRANCH IF OK
03900    2194  010560  104002                      HLT+2                   ;ANS2 NOT EQUAL TO 104000
04000    2195  010562  000340                      340                     ;THE ERROR NUMBER IS 340
04100    2196
04200    2197  010564  022767  100401  167650      CMP     #100401,ANS3    ;CHECK FIRST HALF OF ANSWER
04300    2198  010572  001402                      BEQ     .+6             ;BRANCH IF OK
04400    2199  010574  104004                      HLT+4                   ;ANS3 NOT EQUAL TO 100401
04500    2200  010576  000341                      341                     ;THE ERROR NUMBER IS 341
04600    2201
04700    2202  010600  005767  167640              TST     ANS4            ;CHECK SECOND HALF OF ANSWER
04800    2203  010604  001402                      BEQ     .+6             ;BRANCH IF OK
04900    2204  010606  104004                      HLT+4                   ;ANS4 NOT EQUAL TO 000000
05000    2205  010610  000342                      342                     ;THE ERROR NUMBER IS 342
05100    2206
05200    2207  010612  122767  000047  167564  END47:  CMPB    #47,    $TESTN  ;CHECK THE TEST NUMBER
05300    2208  010620  001402                      BEQ     .+6             ;BRANCH IF OK
05400    2209  010622  104000                      HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
05500    2210  010624  000343                      343                     ;THE ERROR NUMBER IS 343
05600    2211
05700    2212
```

```
00100
00200
00300
00400    2213
00500    2214
00600    2215
00700    2216
00800    2217
00900    2218
01000    2219
01100    2220  010626  104400                ;*********************************************************************
01200    2221  010630  004567  005524        ;TEST 50:      FMUL (LSI-11 FLOATING MULTIPLY INSTRUCTION)
01300    2222  010634  010660                ;       134600,073601 * 104000,104000 = 000401,000000
01400    2223  010636  104000  104000        ;       PS = 200,       STACK POINTER = PC
01500    2224  010642  134600  073601        ;*********************************************************************
01600    2225  010646  000246
01700    2226  010650  016456  000340        TST50:  SCOPE
01800    2227                                        JSR     R5,     PUSH7   ;PUSH 4 WORDS ONTO STACK, SET PRIORITY
01900    2228  010654  000240                        .WORD   STK50           ;TOP OF STACK
02000    2229  010656  075027                        .WORD   104000,104000   ;SECOND OPERAND ON TOP
02100    2230  010660  104000                        .WORD   134600,073601   ;FIRST OPERAND ON BOTTOM
02200    2231  010662  104000                        .WORD   246             ;PROCESSOR PRIORITY LEVEL
02300    2232  010664  134600                        .WORD   TRAPER,340      ;FIS TRAP VECTOR
02400    2233  010666  073601
02500    2234                                        NOP
02600    2235  010670  004767  005514                FMUL    PC              ;FLOATING MULTIPLY ON FOLLOWING 4 WORDS
02700    2236  010674  122767  000200  167530  STK50:  104000          ;SHOULD CONTAIN 104000
02800    2237  010702  001402                        104000          ;SHOULD CONTAIN 104000
02900    2238  010704  104000                        134600          ;BEFORE FMUL, 134600; AFTER, 000401
03000    2239  010706  000344                        073601          ;BEFORE FMUL, 073601; AFTER, 000000
03100    2240
03200    2241  010710  022767  104000  167520        JSR     PC,     POP7    ;POP THE ANSWER
03300    2242  010716  001402                        CMPB    #200,   $PSW    ;CHECK PS (EXCEPT T BIT)
03400    2243  010720  104002                        BEQ     .+6             ;BRANCH IF OK
03500    2244  010722  000345                        HLT                     ;PS NOT EQUAL TO 200
03600    2245                                        344                     ;THE ERROR NUMBER IS 344
03700    2246  010724  022767  104000  167506
03800    2247  010732  001402                        CMP     #104000,ANS1    ;CHECK FIRST HALF OF INPUT DATA (STK50)
03900    2248  010734  104002                        BEQ     .+6             ;BRANCH IF OK
04000    2249  010736  000346                        HLT+2                   ;ANS1 NOT EQUAL TO 104000
04100    2250                                        345                     ;THE ERROR NUMBER IS 345
04200    2251  010740  022767  000401  167474
04300    2252  010746  001402                        CMP     #104000,ANS2    ;CHECK SECOND HALF OF INPUT DATA (STK50+2)
04400    2253  010750  104004                        BEQ     .+6             ;BRANCH IF OK
04500    2254  010752  000347                        HLT+2                   ;ANS2 NOT EQUAL TO 104000
04600    2255                                        346                     ;THE ERROR NUMBER IS 346
04700    2256  010754  005767  167464
04800    2257  010760  001402                        CMP     #000401,ANS3    ;CHECK FIRST HALF OF ANSWER
04900    2258  010762  104004                        BEQ     .+6             ;BRANCH IF OK
05000    2259  010764  000350                        HLT+4                   ;ANS3 NOT EQUAL TO 000401
05100    2260                                        347                     ;THE ERROR NUMBER IS 347
05200    2261  010766  122767  000050  167410
05300    2262  010774  001402                        TST     ANS4            ;CHECK SECOND HALF OF ANSWER
05400    2263  010776  104000                        BEQ     .+6             ;BRANCH IF OK
05500    2264  011000  000351                        HLT+4                   ;ANS4 NOT EQUAL TO 000000
05600    2265                                        350                     ;THE ERROR NUMBER IS 350
05700    2266
05800    2267    END50:  CMPB    #50,    $TESTN  ;CHECK THE TEST NUMBER
05900    2268            BEQ     .+6             ;BRANCH IF OK
                         HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
                         351                     ;THE ERROR NUMBER IS 351

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

```
00300
00400   2269                                    ;TEST 51:    FMUL (LSI-11 FLOATING MULTIPLY INSTRUCTION)
00500   2270                                    ;       024252,125252 * 114100,000000 ==> UNDERFLOW
00600   2271                                    ;       PS(ON STACK) = 212,     STACK POINTER = R0
00700   2272                                    ;*********************************************************************
00800   2273
00900   2274  011002  104400                        SCOPE
01000   2275  011004  004567  005176          TST51: JSR     R5,     PUSHR    ;PUSH 4 WORDS ONTO R0 STACK, SET PRIORITY
01100   2276  011010  114100  000000                .WORD   114100,000000    ;SECOND OPERAND ON TOP
01200   2277  011014  024252  125252                .WORD   024252,125252    ;FIRST OPERAND ON BOTTOM
01300   2278  011020  000305                        .WORD   305              ;PROCESSOR PRIORITY LEVEL
01400   2279  011022  011054  000057                .WORD   ISR51,  057      ;FIS TRAP VECTOR
01500   2280  011026  012700  000510                MOV     #STACK0,R0       ;SET UP R0 AS STACK POINTER
01600   2281
01700   2282  011032  000240                        NOP
01800   2283  011034  075020                        FMUL    R0               ;FLOATING MULTIPLY ON THE R0 STACK
01900   2284
02000   2285  011036  004767  005176          RTA51: JSR     PC,     POPR     ;POP THE ''ANSWER''
02100   2286  011042  010067  167366                MOV     R0,     $SP      ;SAVE STACK POINTER (R0)
02200   2287  011046  104002                        HLT+2                    ;FIS TRAP DIDN'T OCCURE!
02300   2288  011050  000352                        352                      ;THE ERROR NUMBER IS 352
02400   2289  011052  000463                        BR      END51
02500   2290
02600   2291  011054  004767  005210          ISR51: JSR     PC,     POPER    ;POP ALL DATA OFF THE STACKS
02700   2292  011060  010067  167350                MOV     R0,     $SP      ;SAVE STACK POINTER (R0)
02800   2293  011064  122767  000057  167340        CMPB    #057,   $PSW     ;CHECK PS AFTER FIS TRAP
02900   2294  011072  001402                        BEQ     .+6              ;BRANCH IF OK
03000   2295  011074  104000                        HLT                      ;PS AFTER FIS TRAP NOT EQUAL TO 057
03100   2296  011076  000353                        353                      ;THE ERROR NUMBER IS 353
03200   2297
03300   2298  011100  022767  000510  167326        CMP     #STACK0,$SP      ;CHECK THE STACK POINTER (R0)
03400   2299  011106  001402                        BEQ     .+6              ;BRANCH IF OK
03500   2300  011110  104000                        HLT                      ;STACK POINTER (R0) NOT EQUAL TO #STACK0
03600   2301  011112  000354                        354                      ;THE ERROR NUMBER IS 354
03700   2302
03800   2303  011114  022767  011036  167314        CMP     #RTA51, ANS1     ;CHECK FIS TRAP RETURN ADDRESS
03900   2304  011122  001402                        BEQ     .+6              ;BRANCH IF OK
04000   2305  011124  104001                        HLT+1                    ;FIS TRAP AT WRONG ADDRESS
04100   2306  011126  000355                        355                      ;THE ERROR NUMBER IS 355
04200   2307
04300   2308  011130  022767  000212  167302        CMP     #212,   ANS2     ;CHECK PS BEFORE FIS TRAP
04400   2309  011136  001402                        BEQ     .+6              ;BRANCH IF OK
04500   2310  011140  104002                        HLT+2                    ;PS AT FIS TRAP TIME NOT 212
04600   2311  011142  000356                        356                      ;THE ERROR NUMBER IS 356
04700   2312
04800   2313  011144  022767  114100  167270        CMP     #114100,ANS3     ;CHECK DATA FROM THE STACK
04900   2314  011152  001402                        BEQ     .+6              ;BRANCH IF OK
05000   2315  011154  104004                        HLT+4                    ;DATA ON STACK (114100) CHANGED
05100   2316  011156  000357                        357                      ;THE ERROR NUMBER IS 357
05200   2317
05300   2318  011160  005767  167260                TST     ANS4             ;CHECK DATA FROM STACK
05400   2319  011164  001402                        BEQ     .+6              ;BRANCH IF OK
05500   2320  011166  104004                        HLT+4                    ;DATA ON STACK (000000) CHANGED
05600   2321  011170  000360                        360                      ;THE ERROR NUMBER IS 360
05700   2322
05800   2323  011172  022767  024252  167246        CMP     #024252,ANS5     ;CHECK DATA FROM STACK
05900   2324  011200  001402                        BEQ     .+6              ;BRANCH IF OK
```

```
00100    CVKACC  MACY11 30A(1052)  21-AUG-78  15:28  PAGE 55
00200    CVKACC.P11    16-AUG-78 08:41          TEST FLOATING MUL. INSTRUCTION WITH UNDERFLOW                    SEQ 0063
00300
00400     2325  011202  104006                          HLT+6                   ;DATA ON STACK (024252) CHANGED
00500     2326  011204  000361                          361                     ;THE ERROR NUMBER IS 361
00600     2327
00700     2328  011206  022767  125252  167234        CMP     #125252,ANS6    ;CHECK DATA FROM STACK
00800     2329  011214  001402                          BEQ     .+6             ;BRANCH IF OK
00900     2330  011216  104006                          HLT+6                   ;DATA ON STACK (125252) CHANGED
01000     2331  011220  000362                          362                     ;THE ERROR NUMBER IS 362
01100     2332
01200     2333  011222  122767  000051  167154  END51:  CMPB    #51,    $TESTN  ;CHECK THE TEST NUMBER
01300     2334  011230  001402                          BEQ     .+6             ;BRANCH IF OK
01400     2335  011232  104000                          HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
01500     2336  011234  000363                          363                     ;THE ERROR NUMBER IS 363
01600     2337
01700     2338
```

```
00300
00400     2339
00500     2340                    ;*****************************************************************************
00600     2341                    ;TEST 52:       FMUL (LSI-11 FLOATING MULTIPLY INSTRUCTION)
00700     2342                    ;       076452,125252 * 041500,000001 ==> OVERFLOW
00800     2343                    ;       PS(ON STACK) = 002,     STACK POINTER = SP
00900     2344                    ;*****************************************************************************
01000     2345
01100     2346  011236  104400                SCOPE
01200     2347  011240  004567  004570  TST52: JSR     R5,     PUSHS   ;PUSH 4 WORDS ONTO STACK, SET PRIORITY
01300     2348  011244  041500  000001          .WORD   041500,000001   ;SECOND OPERAND ON TOP
01400     2349  011250  076452  125252          .WORD   076452,125252   ;FIRST OPERAND ON BOTTOM
01500     2350  011254  000105                  .WORD   105             ;PROCESSOR PRIORITY LEVEL
01600     2351  011256  011304  000357          .WORD   ISR52,  357     ;FIS TRAP VECTOR
01700     2352
01800     2353  011262  000240                NOP
01900     2354  011264  075026                FMUL    SP              ;FLOATING MULTIPLY ON THE STACK
02000     2355
02100     2356  011266  004767  004602  RTA52: JSR     PC,     POPS    ;POP THE ''ANSWER''
02200     2357  011272  104002                HLT+2                   ;FIS TRAP DIDN'T OCCURE!
02300     2358  011274  000364                364                     ;THE ERROR NUMBER IS 364
02400     2359  011276  012706  000600        MOV     #BEGIN, SP      ;RESTORE THE STACK POINTER
02500     2360  011302  000464                BR      END52
02600     2361
02700     2362  011304  004767  004616  ISR52: JSR     PC,     POPES   ;POP ALL DATA OFF THE STACK
02800     2363  011310  022706  000600        CMP     #BEGIN, SP      ;CHECK THE STACK POINTER
02900     2364  011314  001405                BEQ     ISA52           ;BRANCH IF OK
03000     2365  011316  012706  000600        MOV     #BEGIN, SP      ;RESTORE THE STACK POINTER
03100     2366  011322  104000                HLT                     ;STACK POINTER FOULED UP
03200     2367  011324  000365                365                     ;THE ERROR NUMBER IS 365
03300     2368  011326  000452                BR      END52           ;SKIP REST OF TEST
03400     2369
03500     2370  011330  122767  000357  167074  ISA52: CMPB    #357,   $PSW    ;CHECK PS AFTER FIS TRAP
03600     2371  011336  001402                BEQ     .+6             ;BRANCH IF OK
03700     2372  011340  104000                HLT                     ;PS AFTER FIS TRAP NOT EQUAL TO 357
03800     2373  011342  000366                366                     ;THE ERROR NUMBER IS 366
03900     2374
04000     2375  011344  022767  011266  167064        CMP     #RTA52, ANS1    ;CHECK FIS TRAP RETURN ADDRESS
04100     2376  011352  001402                BEQ     .+6             ;BRANCH IF OK
04200     2377  011354  104001                HLT+1                   ;FIS TRAP AT WRONG ADDRESS
04300     2378  011356  000367                367                     ;THE ERROR NUMBER IS 367
04400     2379
04500     2380  011360  022767  000002  167052        CMP     #002,   ANS2    ;CHECK PS BEFORE FIS TRAP
04600     2381  011366  001402                BEQ     .+6             ;BRANCH IF OK
04700     2382  011370  104002                HLT+2                   ;PS AT FIS TRAP TIME NOT 002
04800     2383  011372  000370                370                     ;THE ERROR NUMBER IS 370
04900     2384
05000     2385  011374  022767  041500  167040        CMP     #041500,ANS3    ;CHECK DATA FROM THE STACK
05100     2386  011402  001402                BEQ     .+6             ;BRANCH IF OK
05200     2387  011404  104004                HLT+4                   ;DATA ON STACK (041500) CHANGED
05300     2388  011406  000371                371                     ;THE ERROR NUMBER IS 371
05400     2389
05500     2390  011410  022767  000001  167026        CMP     #000001,ANS4    ;CHECK DATA FROM STACK
05600     2391  011416  001402                BEQ     .+6             ;BRANCH IF OK
05700     2392  011420  104004                HLT+4                   ;DATA ON STACK (000001) CHANGED
05800     2393  011422  000372                372                     ;THE ERROR NUMBER IS 372
05900     2394
```

```
00100
00200
00300
00400    2395  011424  022767  076452  167014              CMP     #076452,ANS5    ;CHECK DATA FROM STACK
00500    2396  011432  001402                              BEQ     .+6             ;BRANCH IF OK
00600    2397  011434  104006                              HLT+6                   ;DATA ON STACK (076452) CHANGED
00700    2398  011436  000373                              373                     ;THE ERROR NUMBER IS 373
00800    2399
00900    2400  011440  022767  125252  167002              CMP     #125252,ANS6    ;CHECK DATA FROM STACK
01000    2401  011446  001402                              BEQ     .+6             ;BRANCH IF OK
01100    2402  011450  104006                              HLT+6                   ;DATA ON STACK (125252) CHANGED
01200    2403  011452  000374                              374                     ;THE ERROR NUMBER IS 374
01300    2404
01400    2405  011454  122767  000052  166722    END52:    CMPB    #52,    $TESTN  ;CHECK THE TEST NUMBER
01500    2406  011462  001402                              BEQ     .+6             ;BRANCH IF OK
01600    2407  011464  104000                              HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
01700    2408  011466  000375                              375                     ;THE ERROR NUMBER IS 375
01800    2409
01900    2410
```

```
00300
00400    2411
00500    2412                    ;*****************************************************************************
00600    2413                    ;TEST 53:       FDIV (LSI-11 FLOATING DIVIDE INSTRUCTION)
00700    2414                    ;       167452,125251 / 127652,125252 = 077777,177776
00800    2415                    ;       PS = 000,       STACK POINTER = R0
00900    2416                    ;*****************************************************************************
01000    2417
01100    2418  011470  104400                       SCOPE
01200    2419  011472  004567  004510       TST53:  JSR     R5,     PUSHR   ;PUSH 4 WORDS ONTO R0 STACK, SET PRIORITY
01300    2420  011476  127652  125252               .WORD   127652,125252   ;SECOND OPERAND ON TOP
01400    2421  011502  167452  125251               .WORD   167452,125251   ;FIRST OPERAND ON BOTTOM
01500    2422  011506  000111                       .WORD   111             ;PROCESSOR PRIORITY LEVEL
01600    2423  011510  016456  000340               .WORD   TRAPER, 340     ;FIS TRAP VECTOR
01700    2424  011514  012700  000510               MOV     #STACK0,R0      ;SHECK  STACK POINTER
01800    2425
01900    2426  011520  000240                       NOP
02000    2427  011522  075030                       FDIV    R0              ;FLOATING DIVIDE ON THE R0 STACK
02100    2428
02200    2429  011524  004767  004510               JSR     PC,     POPR    ;POP THE ANSWER
02300    2430  011530  010067  166700               MOV     R0,     $SP     ;SAVE ''STACK POINTER''
02400    2431  011534  105767  166672               TSTB    $PSW            ;CHECK PS (EXCEPT T BIT)
02500    2432  011540  001402                       BEQ     .+6             ;BRANCH IF OK
02600    2433  011542  104000                       HLT                     ;PS NOT EQUAL TO 000
02700    2434  011544  000376                       376                     ;THE ERROR NUMBER IS 376
02800    2435
02900    2436  011546  022767  000514  166660       CMP     #STACK4,$SP     ;CHECK THE STACK POINTER (R0)
03000    2437  011554  001402                       BEQ     .+6             ;BRANCH IF OK
03100    2438  011556  104000                       HLT                     ;STACK POINTER (R0) NOT EQUAL TO #STACK4
03200    2439  011560  000377                       377                     ;THE ERROR NUMBER IS 377
03300    2440
03400    2441  011562  022767  077777  166646       CMP     #077777,ANS1    ;CHECK FIRST HALF OF ANSWER
03500    2442  011570  001402                       BEQ     .+6             ;BRANCH IF OK
03600    2443  011572  104002                       HLT+2                   ;ANS1 NOT EQUAL TO 077777
03700    2444  011574  000400                       400                     ;THE ERROR NUMBER IS 400
03800    2445
03900    2446  011576  022767  177776  166634       CMP     #177776,ANS2    ;CHECK SECOND HALF OF ANSWER
04000    2447  011604  001402                       BEQ     .+6             ;BRANCH IF OK
04100    2448  011606  104002                       HLT+2                   ;ANS2 NOT EQUAL TO 177776
04200    2449  011610  000401                       401                     ;THE ERROR NUMBER IS 401
04300    2450
04400    2451  011612  122767  000053  166564 END53:  CMPB    #53,    $TESTN  ;CHECK THE TEST NUMBER
04500    2452  011620  001402                       BEQ     .+6             ;BRANCH IF OK
04600    2453  011622  104000                       HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
04700    2454  011624  000402                       402                     ;THE ERROR NUMBER IS 402
04800    2455
04900    2456
```

```
00300
00400    2457
00500    2458                    ;*********************************************************************************
00600    2459                    ;TEST 54:       FDIV (LSI-11 FLOATING DIVIDE INSTRUCTION)
00700    2460                    ;       167452,125252 / 027652,125253 = 177777,177777
00800    2461                    ;       PS = 210,       STACK POINTER = SP
00900    2462                    ;*********************************************************************************
01000    2463
01100    2464  011626  104400                      SCOPE
01200    2465  011630  004567  004200    TST54:    JSR     R5,     PUSHS   ;PUSH 4 WORDS ONTO STACK, SET PRIORITY
01300    2466  011634  027652  125253              .WORD   027652,125253   ;SECOND OPERAND ON TOP
01400    2467  011640  167452  125252              .WORD   167452,125252   ;FIRST OPERAND ON BOTTOM
01500    2468  011644  000300                      .WORD   300             ;PROCESSOR PRIORITY LEVEL
01600    2469  011646  016456  000340              .WORD   TRAPER, 340     ;FIS TRAP VECTOR
01700    2470
01800    2471  011652  000240                      NOP
01900    2472  011654  075036                      FDIV    SP              ;FLOATING DIVIDE ON THE STACK
02000    2473
02100    2474  011656  004767  004212              JSR     PC,     POPS    ;POP THE ANSWER
02200    2475  011662  022706  000600              CMP     #BEGIN, SP      ;CHECK THE STACK POINTER
02300    2476  011666  001405                      BEQ     TSA54           ;BRANCH IF OK
02400    2477  011670  012706  000600              MOV     #BEGIN, SP      ;RESTORE STACK POINTER
02500    2478  011674  104000                      HLT                     ;STACK POINTER FOULED UP
02600    2479  011676  000403                      403                     ;THE ERROR NUMBER IS 403
02700    2480  011700  000422                      BR      END54           ;SKIP REST OF TEST
02800    2481
02900    2482  011702  122767  000210  166522    TSA54:    CMPB    #210,   $PSW    ;CHECK PS (EXCEPT T BIT)
03000    2483  011710  001402                      BEQ     .+6             ;BRANCH IF OK
03100    2484  011712  104000                      HLT                     ;PS NOT EQUAL TO 210
03200    2485  011714  000404                      404                     ;THE ERROR NUMBER IS 404
03300    2486
03400    2487  011716  022767  177777  166512              CMP     #177777,ANS1    ;CHECK FIRST HALF OF ANSWER
03500    2488  011724  001402                      BEQ     .+6             ;BRANCH IF OK
03600    2489  011726  104002                      HLT+2                   ;ANS1 NOT EQUAL TO 177777
03700    2490  011730  000405                      405                     ;THE ERROR NUMBER IS 405
03800    2491
03900    2492  011732  022767  177777  166500              CMP     #177777,ANS2    ;CHECK SECOND HALF OF ANSWER
04000    2493  011740  001402                      BEQ     .+6             ;BRANCH IF OK
04100    2494  011742  104002                      HLT+2                   ;ANS2 NOT EQUAL TO 177777
04200    2495  011744  000406                      406                     ;THE ERROR NUMBER IS 406
04300    2496
04400    2497  011746  122767  000054  166430    END54:    CMPB    #54,    $TESTN  ;CHECK THE TEST NUMBER
04500    2498  011754  001402                      BEQ     .+6             ;BRANCH IF OK
04600    2499  011756  104000                      HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
04700    2500  011760  000407                      407                     ;THE ERROR NUMBER IS 407
04800    2501
04900    2502
```

```
00300
00400   2503
00500   2504                    ;*****************************************************************************
00600   2505                    ;TEST 55:       FDIV (LSI-11 FLOATING DIVIDE INSTRUCTION)
00700   2506                    ;       125252,125252 / 065252,125252 = 100200,000000
00800   2507                    ;       PS = 210,       STACK POINTER = R2
00900   2508                    ;*****************************************************************************
01000   2509
01100   2510  011762  104400              SCOPE
01200   2511  011764  004567  004216  TST55: JSR     R5,     PUSHR   ;PUSH 4 WORDS ONTO R2 STACK, SET PRIORITY
01300   2512  011770  065252  125252          .WORD   065252,125252   ;SECOND OPERAND ON TOP
01400   2513  011774  125252  125252          .WORD   125252,125252   ;FIRST OPERAND ON BOTTOM
01500   2514  012000  000217                  .WORD   217             ;PROCESSOR PRIORITY LEVEL
01600   2515  012002  016456  000340          .WORD   TRAPER, 340     ;FIS TRAP VECTOR
01700   2516  012006  012702  000510          MOV     #STACK0,R2      ;SHECK  STACK POINTER
01800   2517
01900   2518  012012  000240                  NOP
02000   2519  012014  075032                  FDIV    R2              ;FLOATING DIVIDE ON THE R2 STACK
02100   2520
02200   2521  012016  004767  004216          JSR     PC,     POPR    ;POP THE ANSWER
02300   2522  012022  010267  166406          MOV     R2,     $SP     ;SAVE ''STACK POINTER''
02400   2523  012026  122767  000210  166376  CMPB    #210,   $PSW    ;CHECK PS (EXCEPT T BIT)
02500   2524  012034  001402                  BEQ     .+6             ;BRANCH IF OK
02600   2525  012036  104000                  HLT                     ;PS NOT EQUAL TO 210
02700   2526  012040  000410                  410                     ;THE ERROR NUMBER IS 410
02800   2527
02900   2528  012042  022767  000514  166364  CMP     #STACK4,$SP     ;CHECK THE STACK POINTER (R2)
03000   2529  012050  001402                  BEQ     .+6             ;BRANCH IF OK
03100   2530  012052  104000                  HLT                     ;STACK POINTER (R2) NOT EQUAL TO #STACK4
03200   2531  012054  000411                  411                     ;THE ERROR NUMBER IS 411
03300   2532
03400   2533  012056  022767  100200  166352  CMP     #100200,ANS1    ;CHECK FIRST HALF OF ANSWER
03500   2534  012064  001402                  BEQ     .+6             ;BRANCH IF OK
03600   2535  012066  104002                  HLT+2                   ;ANS1 NOT EQUAL TO 100200
03700   2536  012070  000412                  412                     ;THE ERROR NUMBER IS 412
03800   2537
03900   2538  012072  005767  166342          TST     ANS2            ;CHECK SECOND HALF OF ANSWER
04000   2539  012076  001402                  BEQ     .+6             ;BRANCH IF OK
04100   2540  012100  104002                  HLT+2                   ;ANS2 NOT EQUAL TO 000000
04200   2541  012102  000413                  413                     ;THE ERROR NUMBER IS 413
04300   2542
04400   2543  012104  122767  000055  166272  END55:  CMPB    #55,    $TESTN  ;CHECK THE TEST NUMBER
04500   2544  012112  001402                  BEQ     .+6             ;BRANCH IF OK
04600   2545  012114  104000                  HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
04700   2546  012116  000414                  414                     ;THE ERROR NUMBER IS 414
04800   2547
04900   2548
```

```
00100
00200
00300
00400    2549
00500    2550
00600    2551
00700    2552
00800    2553
00900    2554
01000    2555
01100    2556  012120  104400
01200    2557  012122  004567  004060
01300    2558  012126  140670  123456
01400    2559  012132  000000  000000
01500    2560  012136  000105
01600    2561  012140  016456  000340
01700    2562  012144  012703  000510
01800    2563
01900    2564  012150  000240
02000    2565  012152  075033
02100    2566
02200    2567  012154  004767  004060
02300    2568  012160  010367  166250
02400    2569  012164  122767  000004  166240
02500    2570  012172  001402
02600    2571  012174  104000
02700    2572  012176  000415
02800    2573
02900    2574  012200  022767  000514  166226
03000    2575  012206  001402
03100    2576  012210  104000
03200    2577  012212  000416
03300    2578
03400    2579  012214  005767  166216
03500    2580  012220  001402
03600    2581  012222  104002
03700    2582  012224  000417
03800    2583
03900    2584  012226  005767  166206
04000    2585  012232  001402
04100    2586  012234  104002
04200    2587  012236  000420
04300    2588
04400    2589  012240  122767  000056  166136
04500    2590  012246  001402
04600    2591  012250  104000
04700    2592  012252  000421
04800    2593
04900    2594

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

        SCOPE
TST56:  JSR     R5,     PUSHR   ;PUSH 4 WORDS ONTO R3 STACK, SET PRIORITY
        .WORD   140670,123456   ;SECOND OPERAND ON TOP
        .WORD   000000,000000   ;FIRST OPERAND ON BOTTOM
        .WORD   105             ;PROCESSOR PRIORITY LEVEL
        .WORD   TRAPER, 340     ;FIS TRAP VECTOR
        MOV     #STACK0,R3      ;SHECK  STACK POINTER

        NOP
        FDIV    R3              ;FLOATING DIVIDE ON THE R3 STACK

        JSR     PC,     POPR    ;POP THE ANSWER
        MOV     R3,     $SP     ;SAVE ''STACK POINTER''
        CMPB    #004,   $PSW    ;CHECK PS (EXCEPT T BIT)
        BEQ     .+6             ;BRANCH IF OK
        HLT                     ;PS NOT EQUAL TO 004
        415                     ;THE ERROR NUMBER IS 415

        CMP     #STACK4,$SP     ;CHECK THE STACK POINTER (R3)
        BEQ     .+6             ;BRANCH IF OK
        HLT                     ;STACK POINTER (R3) NOT EQUAL TO #STACK4
        416                     ;THE ERROR NUMBER IS 416

        TST     ANS1            ;CHECK FIRST HALF OF ANSWER
        BEQ     .+6             ;BRANCH IF OK
        HLT+2                   ;ANS1 NOT EQUAL TO 000000
        417                     ;THE ERROR NUMBER IS 417

        TST     ANS2            ;CHECK SECOND HALF OF ANSWER
        BEQ     .+6             ;BRANCH IF OK
        HLT+2                   ;ANS2 NOT EQUAL TO 000000
        420                     ;THE ERROR NUMBER IS 420

END56:  CMPB    #56.    $TESTN  ;CHECK THE TEST NUMBER
        BEQ     .+6             ;BRANCH IF OK
        HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
        421                     ;THE ERROR NUMBER IS 421
```

```
00300
00400    2595
00500    2596                    ;*****************************************************************************
00600    2597                    ;TEST 57:      FDIV (LSI-11 FLOATING DIVIDE INSTRUCTION)
00700    2598                    ;       102500,146000 / 104000,104000 = 036700,000000
00800    2599                    ;       PS = 200,       STACK POINTER = PC
00900    2600                    ;*****************************************************************************
01000    2601
01100    2602  012254  104400            SCOPE
01200    2603  012256  004567  004076  TST57: JSR     R5,     PUSH7   ;PUSH 4 WORDS ONTO STACK, SET PRIORITY
01300    2604  012262  012306            .WORD   STK57           ;TOP OF STACK
01400    2605  012264  104000  104000    .WORD   104000,104000   ;SECOND OPERAND ON TOP
01500    2606  012270  102500  146000    .WORD   102500,146000   ;FIRST OPERAND ON BOTTOM
01600    2607  012274  000357            .WORD   357             ;PROCESSOR PRIORITY LEVEL
01700    2608  012276  016456  000340    .WORD   TRAPER,340      ;FIS TRAP VECTOR
01800    2609
01900    2610  012302  000240            NOP
02000    2611  012304  075037            FDIV    PC              ;FLOATING DIVIDE ON FOLLOWING 4 WORDS
02100    2612  012306  104000    STK57:  104000                  ;SHOULD CONTAIN 104000
02200    2613  012310  104000            104000                  ;SHOULD CONTAIN 104000
02300    2614  012312  102500            102500                  ;BEFORE FDIV, 102500; AFTER, 036700
02400    2615  012314  146000            146000                  ;BEFORE FDIV, 146000; AFTER, 000000
02500    2616
02600    2617  012316  004767  004066            JSR     PC,     POP7    ;POP THE ANSWER
02700    2618  012322  122767  000200  166102    CMPB    #200,   $PSW    ;CHECK PS (EXCEPT T BIT)
02800    2619  012330  001402            BEQ     .+6             ;BRANCH IF OK
02900    2620  012332  104000            HLT                     ;PS NOT EQUAL TO 200
03000    2621  012334  000422            422                     ;THE ERROR NUMBER IS 422
03100    2622
03200    2623  012336  022767  104000  166072    CMP     #104000,ANS1    ;CHECK FIRST HALF OF INPUT DATA (STK57)
03300    2624  012344  001402            BEQ     .+6             ;BRANCH IF OK
03400    2625  012346  104002            HLT+2                   ;ANS1 NOT EQUAL TO 104000
03500    2626  012350  000423            423                     ;THE ERROR NUMBER IS 423
03600    2627
03700    2628  012352  022767  104000  166060    CMP     #104000,ANS2    ;CHECK SECOND HALF OF INPUT DATA (STK57+2)
03800    2629  012360  001402            BEQ     .+6             ;BRANCH IF OK
03900    2630  012362  104002            HLT+2                   ;ANS2 NOT EQUAL TO 104000
04000    2631  012364  000424            424                     ;THE ERROR NUMBER IS 424
04100    2632
04200    2633  012366  022767  036700  166046    CMP     #036700,ANS3    ;CHECK FIRST HALF OF ANSWER
04300    2634  012374  001402            BEQ     .+6             ;BRANCH IF OK
04400    2635  012376  104004            HLT+4                   ;ANS3 NOT EQUAL TO 036700
04500    2636  012400  000425            425                     ;THE ERROR NUMBER IS 425
04600    2637
04700    2638  012402  005767  166036            TST     ANS4            ;CHECK SECOND HALF OF ANSWER
04800    2639  012406  001402            BEQ     .+6             ;BRANCH IF OK
04900    2640  012410  104004            HLT+4                   ;ANS4 NOT EQUAL TO 000000
05000    2641  012412  000426            426                     ;THE ERROR NUMBER IS 426
05100    2642
05200    2643  012414  122767  000057  165762  END57:  CMPB    #57,    $TESTN  ;CHECK THE TEST NUMBER
05300    2644  012422  001402            BEQ     .+6             ;BRANCH IF OK
05400    2645  012424  104000            HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
05500    2646  012426  000427            427                     ;THE ERROR NUMBER IS 427
05600    2647
05700    2648
05800    2649
05900    2650                    ;*****************************************************************************
```

```
00100
00200
00300
00400    2651                                   ;TEST 60:       FDIV (LSI-11 FLOATING DIVIDE INSTRUCTION)
00500    2652                                   ;       025252,125251 / 065252,125252 ==> UNDERFLOW
00600    2653                                   ;       PS(ON STACK) = 012,     STACK POINTER = R1
00700    2654                                   ;*********************************************************************
00800    2655
00900    2656  012430  104400                              SCOPE
01000    2657  012432  004567  003550           TST60:  JSR     R5,     PUSHR   ;PUSH 4 WORDS ONTO R1 STACK, SET PRIORITY
01100    2658  012436  065252  125252                   .WORD   065252,125252   ;SECOND OPERAND ON TOP
01200    2659  012442  025252  125251                   .WORD   025252,125251   ;FIRST OPERAND ON BOTTOM
01300    2660  012446  000015                           .WORD   015             ;PROCESSOR PRIORITY LEVEL
01400    2661  012450  012502  000300                   .WORD   ISR60,  300     ;FIS TRAP VECTOR
01500    2662  012454  012701  000510                   MOV     #STACK0,R1      ;SET UP R1 AS STACK POINTER
01600    2663
01700    2664  012460  000240                           NOP
01800    2665  012462  075031                           FDIV    R1              ;FLOATING DIVIDE ON THE R1 STACK
01900    2666
02000    2667  012464  004767  003550           RTA60:  JSR     PC,     POPR    ;POP THE ''ANSWER''
02100    2668  012470  010167  165740                   MOV     R1,     $SP     ;SAVE STACK POINTER (R1)
02200    2669  012474  104002                           HLT+2                   ;FIS TRAP DIDN'T OCCURE!
02300    2670  012476  000430                           430                     ;THE ERROR NUMBER IS 430
02400    2671  012500  000464                           BR      END60
02500    2672
02600    2673  012502  004767  003562           ISR60:  JSR     PC,     POPER   ;POP ALL DATA OFF THE STACKS
02700    2674  012506  010167  165722                   MOV     R1,     $SP     ;SAVE STACK POINTER (R1)
02800    2675  012512  122767  000300  165712           CMPB    #300,   $PSW    ;CHECK PS AFTER FIS TRAP
02900    2676  012520  001402                           BEQ     .+6             ;BRANCH IF OK
03000    2677  012522  104000                           HLT                     ;PS AFTER FIS TRAP NOT EQUAL TO 300
03100    2678  012524  000431                           431                     ;THE ERROR NUMBER IS 431
03200    2679
03300    2680  012526  022767  000510  165700         CMP     #STACK0,$SP     ;CHECK THE STACK POINTER (R1)
03400    2681  012534  001402                           BEQ     .+6             ;BRANCH IF OK
03500    2682  012536  104000                           HLT                     ;STACK POINTER (R1) NOT EQUAL TO #STACK0
03600    2683  012540  000432                           432                     ;THE ERROR NUMBER IS 432
03700    2684
03800    2685  012542  022767  012464  165666         CMP     #RTA60, ANS1    ;CHECK FIS TRAP RETURN ADDRESS
03900    2686  012550  001402                           BEQ     .+6             ;BRANCH IF OK
04000    2687  012552  104001                           HLT+1                   ;FIS TRAP AT WRONG ADDRESS
04100    2688  012554  000433                           433                     ;THE ERROR NUMBER IS 433
04200    2689
04300    2690  012556  022767  000012  165654         CMP     #012,   ANS2    ;CHECK PS BEFORE FIS TRAP
04400    2691  012564  001402                           BEQ     .+6             ;BRANCH IF OK
04500    2692  012566  104002                           HLT+2                   ;PS AT FIS TRAP TIME NOT 012
04600    2693  012570  000434                           434                     ;THE ERROR NUMBER IS 434
04700    2694
04800    2695  012572  022767  065252  165642         CMP     #065252,ANS3    ;CHECK DATA FROM THE STACK
04900    2696  012600  001402                           BEQ     .+6             ;BRANCH IF OK
05000    2697  012602  104004                           HLT+4                   ;DATA ON STACK (065252) CHANGED
05100    2698  012604  000435                           435                     ;THE ERROR NUMBER IS 435
05200    2699
05300    2700  012606  022767  125252  165630         CMP     #125252,ANS4    ;CHECK DATA FROM STACK
05400    2701  012614  001402                           BEQ     .+6             ;BRANCH IF OK
05500    2702  012616  104004                           HLT+4                   ;DATA ON STACK (125252) CHANGED
05600    2703  012620  000436                           436                     ;THE ERROR NUMBER IS 436
05700    2704
05800    2705  012622  022767  025252  165616         CMP     #025252,ANS5    ;CHECK DATA FROM STACK
05900    2706  012630  001402                           BEQ     .+6             ;BRANCH IF OK
```

```
00100   CVKACC  MACY11 30A(1052)  21-AUG-78  15:28  PAGE 64
00200   CVKACC.P11    16-AUG-78 08:41          TEST FLOATING DIV. INSTRUCTION WITH UNDERFLOW                    SEQ 0072
00300
00400     2707  012632  104006,                                 HLT+6                   ;DATA ON STACK (025252) CHANGED
00500     2708  012634  000437                                  437                     ;THE ERROR NUMBER IS 437
00600     2709
00700     2710  012636  022767  125251  165604                  CMP     #125251,ANS6    ;CHECK DATA FROM STACK
00800     2711  012644  001402                                  BEQ     .+6             ;BRANCH IF OK
00900     2712  012646  104006                                  HLT+6                   ;DATA ON STACK (125251) CHANGED
01000     2713  012650  000440                                  440                     ;THE ERROR NUMBER IS 440
01100     2714
01200     2715  012652  122767  000060  165524          END60:  CMPB    #60,    $TESTN  ;CHECK THE TEST NUMBER
01300     2716  012660  001402                                  BEQ     .+6             ;BRANCH IF OK
01400     2717  012662  104000                                  HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
01500     2718  012664  000441                                  441                     ;THE ERROR NUMBER IS 441
01600     2719
01700     2720
```

```
00300
00400    2721
00500    2722                    ;*****************************************************************************
00600    2723                    ;TEST #1:      FDIV (LSI-11 FLOATING DIVIDE INSTRUCTION)
00700    2724                    ;       067452,125252 / 127652,125252 ==> OVERFLOW
00800    2725                    ;       PS(ON STACK) = 202,     STACK POINTER = R4
00900    2726                    ;*****************************************************************************
01000    2727
01100    2728  012666  104400            SCOPE
01200    2729  012670  004567  003312 TST61: JSR     R5,     PUSHR   ;PUSH 4 WORDS ONTO R4 STACK, SET PRIORITY
01300    2730  012674  127652  125252       .WORD   127652,125252   ;SECOND OPERAND ON TOP
01400    2731  012700  067452  125252       .WORD   067452,125252   ;FIRST OPERAND ON BOTTOM
01500    2732  012704  000242               .WORD   242             ;PROCESSOR PRIORITY LEVEL
01600    2733  012706  012740  000357       .WORD   ISR61, 357      ;FIS TRAP VECTOR
01700    2734  012712  012704  000510       MOV     #STACK0,R4      ;SET UP R4 AS STACK POINTER
01800    2735
01900    2736  012716  000240               NOP
02000    2737  012720  075034               FDIV    R4              ;FLOATING DIVIDE ON THE R4 STACK
02100    2738
02200    2739  012722  004767  003312 RTA61: JSR     PC,     POPR    ;POP THE ''ANSWER''
02300    2740  012726  010467  165502       MOV     R4,     $SP     ;SAVE STACK POINTER (R4)
02400    2741  012732  104002               HLT+2                   ;FIS TRAP DIDN'T OCCURE!
02500    2742  012734  000442               442                     ;THE ERROR NUMBER IS 442
02600    2743  012736  000464               BR      END61
02700    2744
02800    2745  012740  004767  003324 ISR61: JSR     PC,     POPER   ;POP ALL DATA OFF THE STACKS
02900    2746  012744  010467  165464       MOV     R4,     $SP     ;SAVE STACK POINTER (R4)
03000    2747  012750  122767  000357  165454  CMPB    #357,   $PSW    ;CHECK PS AFTER FIS TRAP
03100    2748  012756  001402               BEQ     .+6             ;BRANCH IF OK
03200    2749  012760  104000               HLT                     ;PS AFTER FIS TRAP NOT EQUAL TO 357
03300    2750  012762  000443               443                     ;THE ERROR NUMBER IS 443
03400    2751
03500    2752  012764  022767  000510  165442  CMP     #STACK0,$SP     ;CHECK THE STACK POINTER (R4)
03600    2753  012772  001402               BEQ     .+6             ;BRANCH IF OK
03700    2754  012774  104000               HLT                     ;STACK POINTER (R4) NOT EQUAL TO #STACK0
03800    2755  012776  000444               444                     ;THE ERROR NUMBER IS 444
03900    2756
04000    2757  013000  022767  012722  165430  CMP     #RTA61, ANS1    ;CHECK FIS TRAP RETURN ADDRESS
04100    2758  013006  001402               BEQ     .+6             ;BRANCH IF OK
04200    2759  013010  104001               HLT+1                   ;FIS TRAP AT WRONG ADDRESS
04300    2760  013012  000445               445                     ;THE ERROR NUMBER IS 445
04400    2761
04500    2762  013014  022767  000202  165416  CMP     #202,   ANS2    ;CHECK PS BEFORE FIS TRAP
04600    2763  013022  001402               BEQ     .+6             ;BRANCH IF OK
04700    2764  013024  104002               HLT+2                   ;PS AT FIS TRAP TIME NOT 202
04800    2765  013026  000446               446                     ;THE ERROR NUMBER IS 446
04900    2766
05000    2767  013030  022767  127652  165404  CMP     #127652,ANS3    ;CHECK DATA FROM THE STACK
05100    2768  013036  001402               BEQ     .+6             ;BRANCH IF OK
05200    2769  013040  104004               HLT+4                   ;DATA ON STACK (127652) CHANGED
05300    2770  013042  000447               447                     ;THE ERROR NUMBER IS 447
05400    2771
05500    2772  013044  022767  125252  165372  CMP     #125252,ANS4    ;CHECK DATA FROM STACK
05600    2773  013052  001402               BEQ     .+6             ;BRANCH IF OK
05700    2774  013054  104004               HLT+4                   ;DATA ON STACK (125252) CHANGED
05800    2775  013056  000450               450                     ;THE ERROR NUMBER IS 450
05900    2776
```

```
00100
00200
00300
00400    2777  013060  022767  067452  165360            CMP     #067452,ANS5    ;CHECK DATA FROM STACK
00500    2778  013066  001402                            BEQ     .+6             ;BRANCH IF OK
00600    2779  013070  104006                            HLT+6                   ;DATA ON STACK (067452) CHANGED
00700    2780  013072  000451                            451                     ;THE ERROR NUMBER IS 451
00800    2781
00900    2782  013074  022767  125252  165346            CMP     #125252,ANS6    ;CHECK DATA FROM STACK
01000    2783  013102  001402                            BEQ     .+6             ;BRANCH IF OK
01100    2784  013104  104006                            HLT+6                   ;DATA ON STACK (125252) CHANGED
01200    2785  013106  000452                            452                     ;THE ERROR NUMBER IS 452
01300    2786
01400    2787  013110  122767  000061  165266  END61:    CMPB    #61.    $TESTN  ;CHECK THE TEST NUMBER
01500    2788  013116  001402                            BEQ     .+6             ;BRANCH IF OK
01600    2789  013120  104000                            HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
01700    2790  013122  000453                            453                     ;THE ERROR NUMBER IS 453
01800    2791
01900    2792
```

```
00300
00400    2793
00500    2794                    ;*************************************************************************
00600    2795                    ;TEST 62:       FDIV (LSI-11 FLOATING DIVIDE INSTRUCTION)
00700    2796                    ;       052525,052525 / 100125,125252 ==> DIVIDE BY ZERO
00800    2797                    ;       PS(ON STACK) = 013,     STACK POINTER = R5
00900    2798                    ;*************************************************************************
01000    2799
01100    2800  013124  104400              SCOPE
01200    2801  013126  004567  003054  TST62: JSR     R5,     PUSHR   ;PUSH 4 WORDS ONTO R5 STACK, SET PRIORITY
01300    2802  013132  100125  125252        .WORD   100125,125252   ;SECOND OPERAND ON TOP
01400    2803  013136  052525  052525        .WORD   052525,052525   ;FIRST OPERAND ON BOTTOM
01500    2804  013142  000047                .WORD   047             ;PROCESSOR PRIORITY LEVEL
01600    2805  013144  013176  000113        .WORD   ISR62,  113     ;FIS TRAP VECTOR
01700    2806  013150  012705  000510        MOV     #STACK0,R5      ;SET UP R5 AS STACK POINTER
01800    2807
01900    2808  013154  000240              NOP
02000    2809  013156  075035              FDIV    R5              ;FLOATING DIVIDE ON THE R5 STACK
02100    2810
02200    2811  013160  004767  003054  RTA62: JSR     PC,     POPR    ;POP THE ''ANSWER''
02300    2812  013164  010567  165244        MOV     R5,     $SP     ;SAVE STACK POINTER (R5)
02400    2813  013170  104002                HLT+2                   ;FIS TRAP DIDN'T OCCURE!
02500    2814  013172  000454                454                     ;THE ERROR NUMBER IS 454
02600    2815  013174  000464                BR      END62
02700    2816
02800    2817  013176  004767  003066  ISR62: JSR     PC,     POPER   ;POP ALL DATA OFF THE STACKS
02900    2818  013202  010567  165226        MOV     R5,     $SP     ;SAVE STACK POINTER (R5)
03000    2819  013206  122767  000113  165216  CMPB    #113,   $PSW    ;CHECK PS AFTER FIS TRAP
03100    2820  013214  001402                BEQ     .+6             ;BRANCH IF OK
03200    2821  013216  104000                HLT                     ;PS AFTER FIS TRAP NOT EQUAL TO 113
03300    2822  013220  000455                455                     ;THE ERROR NUMBER IS 455
03400    2823
03500    2824  013222  022767  000510  165204  CMP     #STACK0,$SP     ;CHECK THE STACK POINTER (R5)
03600    2825  013230  001402                BEQ     .+6             ;BRANCH IF OK
03700    2826  013232  104000                HLT                     ;STACK POINTER (R5) NOT EQUAL TO #STACK0
03800    2827  013234  000456                456                     ;THE ERROR NUMBER IS 456
03900    2828
04000    2829  013236  022767  013160  165172  CMP     #RTA62, ANS1    ;CHECK FIS TRAP RETURN ADDRESS
04100    2830  013244  001402                BEQ     .+6             ;BRANCH IF OK
04200    2831  013246  104001                HLT+1                   ;FIS TRAP AT WRONG ADDRESS
04300    2832  013250  000457                457                     ;THE ERROR NUMBER IS 457
04400    2833
04500    2834  013252  022767  000013  165160  CMP     #013,   ANS2    ;CHECK PS BEFORE FIS TRAP
04600    2835  013260  001402                BEQ     .+6             ;BRANCH IF OK
04700    2836  013262  104002                HLT+2                   ;PS AT FIS TRAP TIME NOT 013
04800    2837  013264  000460                460                     ;THE ERROR NUMBER IS 460
04900    2838
05000    2839  013266  022767  100125  165146  CMP     #100125,ANS3    ;CHECK DATA FROM THE STACK
05100    2840  013274  001402                BEQ     .+6             ;BRANCH IF OK
05200    2841  013276  104004                HLT+4                   ;DATA ON STACK (100125) CHANGED
05300    2842  013300  000461                461                     ;THE ERROR NUMBER IS 461
05400    2843
05500    2844  013302  022767  125252  165134  CMP     #125252,ANS4    ;CHECK DATA FROM STACK
05600    2845  013310  001402                BEQ     .+6             ;BRANCH IF OK
05700    2846  013312  104004                HLT+4                   ;DATA ON STACK (125252) CHANGED
05800    2847  013314  000462                462                     ;THE ERROR NUMBER IS 462
05900    2848
```

```
00100   CVKACC  MACY11 30A(1052)  21-AUG-78  15:28  PAGE 68
00200   CVKACC.P11    16-AUG-78 08:41          TEST FLOATING DIV. INSTRUCTION FOR DIVIDE BY ZERO                    SEQ 0076
00300
00400     2849  013316  022767  052525  165122                  CMP     #052525,ANS5    ;CHECK DATA FROM STACK
00500     2850  013324  001402                                  BEQ     .+6             ;BRANCH IF OK
00600     2851  013326  104006                                  HLT+6                   ;DATA ON STACK (052525) CHANGED
00700     2852  013330  000463                                  463                     ;THE ERROR NUMBER IS 463
00800     2853
00900     2854  013332  022767  052525  165110                  CMP     #052525,ANS6    ;CHECK DATA FROM STACK
01000     2855  013340  001402                                  BEQ     .+6             ;BRANCH IF OK
01100     2856  013342  104006                                  HLT+6                   ;DATA ON STACK (052525) CHANGED
01200     2857  013344  000464                                  464                     ;THE ERROR NUMBER IS 464
01300     2858
01400     2859  013346  122767  000062  165030          END62:  CMPB    #62,    $TESTN  ;CHECK THE TEST NUMBER
01500     2860  013354  001402                                  BEQ     .+6             ;BRANCH IF OK
01600     2861  013356  104000                                  HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
01700     2862  013360  000465                                  465                     ;THE ERROR NUMBER IS 465
01800     2863
01900     2864
```

```
00300
00400    2865
00500    2866                    ;*****************************************************************************
00600    2867                    ;TEST 63:       FDIV (LSI-11 FLOATING DIVIDE INSTRUCTION)
00700    2868                    ;       100052,052525 / 000006,123456 ==> DIVIDE BY ZERO
00800    2869                    ;       PS(ON STACK) = 213,     STACK POINTER = SP
00900    2870                    ;*****************************************************************************
01000    2871
01100    2872 013362 104400              SCOPE
01200    2873 013364 004567 002444 TST63: JSR     R5,     PUSHS   ;PUSH 4 WORDS ONTO STACK, SET PRIORITY
01300    2874 013370 000006 123456        .WORD   000006,123456   ;SECOND OPERAND ON TOP
01400    2875 013374 100052 052525        .WORD   100052,052525   ;FIRST OPERAND ON BOTTOM
01500    2876 013400 000357               .WORD   357             ;PROCESSOR PRIORITY LEVEL
01600    2877 013402 013430 000311        .WORD   ISR63,  311     ;FIS TRAP VECTOR
01700    2878
01800    2879 013406 000240              NOP
01900    2880 013410 075036              FDIV    SP              ;FLOATING DIVIDE ON THE STACK
02000    2881
02100    2882 013412 004767 002456 RTA63: JSR     PC,     POPS    ;POP THE ''ANSWER''
02200    2883 013416 104002               HLT+2                  ;FIS TRAP DIDN'T OCCURE!
02300    2884 013420 000466               466                    ;THE ERROR NUMBER IS 466
02400    2885 013422 012706 000600        MOV     #BEGIN, SP      ;RESTORE THE STACK POINTER
02500    2886 013426 000464               BR      END63
02600    2887
02700    2888 013430 004767 002472 ISR63: JSR     PC,     POPES   ;POP ALL DATA OFF THE STACK
02800    2889 013434 022706 000600        CMP     #BEGIN, SP      ;CHECK THE STACK POINTER
02900    2890 013440 001405               BEQ     ISA63           ;BRANCH IF OK
03000    2891 013442 012706 000600        MOV     #BEGIN, SP      ;RESTORE THE STACK POINTER
03100    2892 013446 104000               HLT                     ;STACK POINTER FOULED UP
03200    2893 013450 000467               467                     ;THE ERROR NUMBER IS 467
03300    2894 013452 000452               BR      END63           ;SKIP REST OF TEST
03400    2895
03500    2896 013454 122767 000311 164750 ISA63: CMPB    #311,   $PSW    ;CHECK PS AFTER FIS TRAP
03600    2897 013462 001402               BEQ     .+6             ;BRANCH IF OK
03700    2898 013464 104000               HLT                     ;PS AFTER FIS TRAP NOT EQUAL TO 311
03800    2899 013466 000470               470                     ;THE ERROR NUMBER IS 470
03900    2900
04000    2901 013470 022767 013412 164740        CMP     #RTA63, ANS1    ;CHECK FIS TRAP RETURN ADDRESS
04100    2902 013476 001402               BEQ     .+6             ;BRANCH IF OK
04200    2903 013500 104001               HLT+1                   ;FIS TRAP AT WRONG ADDRESS
04300    2904 013502 000471               471                     ;THE ERROR NUMBER IS 471
04400    2905
04500    2906 013504 022767 000213 164726        CMP     #213,   ANS2    ;CHECK PS BEFORE FIS TRAP
04600    2907
04700    2908
04800    2909
04900    2910
05000    2911
05100    2912
05200    2913 013512 001402               BEQ     .+6             ;BRANCH IF OK
05300    2914 013514 104002               HLT+2                   ;PS AT FIS TRAP TIME NOT 213
05400    2915 013516 000472               472                     ;THE ERROR NUMBER IS 472
05500    2916
05600    2917 013520 022767 000006 164714        CMP     #000006,ANS3    ;CHECK DATA FROM THE STACK
05700    2918 013526 001402               BEQ     .+6             ;BRANCH IF OK
05800    2919 013530 104004               HLT+4                   ;DATA ON STACK (000006) CHANGED
05900    2920 013532 000473               473                     ;THE ERROR NUMBER IS 473
```

```
00100
00200
00300
00400     2921
00500     2922  013534  022767  123456  164702                CMP     #123456,ANS4    ;CHECK DATA FROM STACK
00600     2923  013542  001402                                BEQ     .+6             ;BRANCH IF OK
00700     2924  013544  104004                                HLT+4                   ;DATA ON STACK (123456) CHANGED
00800     2925  013546  000474                                474                     ;THE ERROR NUMBER IS 474
00900     2926
01000     2927  013550  022767  100052  164670                CMP     #100052,ANS5    ;CHECK DATA FROMONG TK
01100     2928  013556  001402                                BEQ     .+6             ;BRANCH IF OK
01200     2929  013560  104006                                HLT+6                   ;DATA ON STACK (100052) CHANGED
01300     2930  013562  000475                                475                     ;THE ERROR NUMBER IS 475
01400     2931
01500     2932  013564  022767  052525  164656                CMP     #052525,ANS6    ;CHECK DATA FROM STACK
01600     2933  013572  001402                                BEQ     .+6             ;BRANCH IF OK
01700     2934. 013574  104006                                HLT+6                   ;DATA ON STACK (052525) CHANGED
01800     2935  013576  000476                                476                     ;THE ERROR NUMBER IS 476
01900     2936
02000     2937  013600  122767  000063  164576  END63:        CMPB    #63,    $TESTN  ;CHECK THE TEST NUMBER
02100     2938  013606  001402                                BEQ     .+6             ;BRANCH IF OK
02200     2939  013610  104000                                HLT                     ;WREST!  PC MUST HAVE FOULED UP.
02300     2940  013612  000477                                477                     ;THE ERROR NUMBER IS 477
02400     2941
02500     2942
```

```
00100
00200
00300
00400    2943
00500    2944
00600    2945
00700    2946
00800    2947
00900    2948
01000    2949
01100    2950
01200    2951
01300    2952  013614  104400
01400    2953  013616  012704  000532
01500    2954  013622  012744  107070
01600    2955  013626  012744  134343
01700    2956  013632  012744  065432
01800    2957  013636  012744  032107
01900    2958  013642  012744  123456
02000    2959  013646  012744  045670
02100    2960  013652  012744  125252
02200    2961  013656  012744  135252
02300    2962  013662  012744  016161
02400    2963  013666  012744  040616
02500    2964  013672
02600    2965  013672  106427
02700    2966
02800    2967  013676  000240
02900    2968  013700  075014
03000    2969  013702  075034
03100    2970  013704  075024
03200    2971  013706  075004
03300    2972
03400    2973  013710
03500    2974  013710  106767
03600    2975  013714  042767  000020  164510
03700    2976  013722  012467  164510
03800    2977  013726  012467  164506
03900    2978  013732  010467  164476
04000    2979  013736  122767  000010  164466
04100    2980  013744  001402
04200    2981  013746  104000
04300    2982  013750  000500
04400    2983
04500    2984  013752  022767  000532  164454
04600    2985  013760  001402
04700    2986  013762  104000
04800    2987  013764  000501
04900    2988
05000    2989
05100    2990  013766  022767  137201  164442
05200    2991  013774  001402
05300    2992  013776  104002
05400    2993  014000  000502
05500    2994
05600    2995  014002  022767  115230  164430
05700    2996  014010  001402
05800    2997  014012  104002
05900    2998  014014  000503

;*****************************************************************************
;TEST 64:       TEST ALL INSTRUCTION TOGETHER
;                                032107,065432 * 045670,123456
;       134343,107070 + ------------------------------------  = 137201,115230
;                        (135252,125252 - 040616,016161)
;       PS=010, STACK POINTER=R4
;*****************************************************************************

        SCOPE
TST64:  MOV     #STAK10,R4      ;SET STACK POINTER
        MOV     #107070,-(R4)   ;LOAD DATA ONTO STACK
        MOV     #134343,-(R4)
        MOV     #065432,-(R4)
        MOV     #032107,-(R4)
        MOV     #123456,-(R4)
        MOV     #045670,-(R4)
        MOV     #125252,-(R4)
        MOV     #135252,-(R4)
        MOV     #016161,-(R4)
        MOV     #040616,-(R4)
        MTPS    #144            ;SET PROCESSOR STATUS
        .WORD   106400!..C

        NOP
        FSUB    R4      ;135252,125252-040616,016161=140616,017434
        FDIV    R4      ;045670,123456/140616,017434=145246,047065
        FMUL    R4      ;032107,065432*145246,047065=137201,106137
        FADD    R4      ;134343,107070+137201,106137=137201,115230

        MFPS    $PSW            ;SAVE FINAL PS
        .WORD   106700!..C
        BIC     #20,    $PSW    ;CLR T-BIT
        MOV     (R4)+,  ANS1    ;SAVE FIRST HALF OF ANSWER
        MOV     (R4)+,  ANS2    ;SAVE SECOND HALF OF ANSWER
        MOV     R4,     $SP     ;SAVE STACK POINTER
        CMPB    #010,   $PSW    ;CHECK PS (EXCEPT T BIT)
        BEQ     .+6             ;BRANCH IF OK
        HLT                     ;PS NOT EQUAL TO 010
        500                     ;THE ERROR NUMBER IS 500

        CMP     #STAK10,$SP     ;CHECK THE STACK POINTER (R4)
        BEQ     .+6             ;BRANCH IF OK
        HLT                     ;STACK POINTER (R4) NOT EQUAL TO THE
        501                     ;THE ERROR NUMBER IS 501
                                ;ADDRESS OF STAK10

        CMP     #137201,ANS1    ;CHECK FIRST HALF OF ANSWER
        BEQ     .+6             ;BRANCH IF OK
        HLT+2                   ;ANS1 NOT EQUAL TO 137201
        502                     ;THE ERROR NUMBER IS 502

        CMP     #115230,ANS2    ;CHECK SECOND HALF OF ANSWER
        BEQ     .+6             ;BRANCH IF OK
        HLT+2                   ;ANS2 NOT EQUAL TO 115230
        503                     ;THE ERROR NUMBER IS 503
```

```
00100    CVKACC  MACY11 30A(1052)  21-AUG-78  15:28  PAGE 72
00200    CVKACC.P11    16-AUG-78 08:41          TEST OF ALL FIS AT ONCE                                    SEQ 0080
00300
00400       2999
00500       3000  014016  122767  000064  164360          END64:  CMPB    #64,    $TESTN  ;CHECK THE TEST NUMBER
00600       3001  014024  001402                                  BEQ     .+6             ;BRANCH IF OK
00700       3002  014026  104000                                  HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
00800       3003  014030  000504                                  504                     ;THE ERROR NUMBER IS 504
00900       3004
```

```
00300
00400    3005
00500    3006                    ;*****************************************************************************
00600    3007                    ;TEST 65:       TEST THAT STACK POINTER ADDRESS ERROR CAUSES ABORT
00700    3008                    ;       INSTRUCTION = FADD,     STACK POINTER = R2
00800    3009                    ;*****************************************************************************
00900    3010
01000    3011  014032  104400             SCOPE
01100    3012  014034  012737  014124  000004  TST65:  MOV     #ISR65, @#4     ;SET UP ADDRESS TRAP VECTOR
01200    3013  014042  012737  000340  000006          MOV     #340,   @#6
01300    3014  014050  004567  002132          JSR     R5,     PUSHR   ;PUSH 4 WORDS ONTO R2 STACK, SET PRIORITY
01400    3015  014054  070707  016161          .WORD   070707,016161   ;SECOND OPERAND ON TOP
01500    3016  014060  146314  143434          .WORD   146314,143434   ;FIRST OPERAND ON BOTTOM
01600    3017  014064  000143                  .WORD   143             ;PROCESSOR PRIORITY LEVEL
01700    3018  014066  016456  000340          .WORD   TRAPER, 340     ;FIS TRAP VECTOR
01800    3019  014072                          MTPS    #143            ;SET PROCESSOR STATUS
01900    3020  014072  106427                  .WORD   106400!..C
02000    3021  014076  012702  177777          MOV     #177777,R2      ;SET UP R2 AS STACK POINTER
02100    3022
02200    3023  014102  000240                  NOP
02300    3024  014104  075002                  FADD    R2              ;FLOATING ADD ON THE R2 STACK
02400    3025
02500    3026  014106  004767  002126  RTA65:  JSR     PC,     POPR    ;POP THE ''ANSWER''
02600    3027  014112  010267  164316          MOV     R2,     $SP     ;SAVE STACK POINTER (R2)
02700    3028  014116  104002                  HLT+2                   ;FIS TRAP DIDN'T OCCURE!
02800    3029  014120  000505                  505                     ;THE ERROR NUMBER IS 505
02900    3030  014122  000464                  BR      END65
03000    3031
03100    3032  014124  004767  002140  ISR65:  JSR     PC,     POPER   ;POP ALL DATA OFF THE STACKS
03200    3033  014130  010267  164300          MOV     R2,     $SP     ;SAVE STACK POINTER (R2)
03300    3034  014134  122767  000340  164270  CMPB    #340,   $PSW    ;CHECK PS AFTER ADR. ERR. TRAP
03400    3035  014142  001402                  BEQ     .+6             ;BRANCH IF OK
03500    3036  014144  104000                  HLT                     ;PS AFTER TRAP NOT EQUAL TO 340
03600    3037  014146  000506                  506                     ;THE ERROR NUMBER IS 506
03700    3038
03800    3039  014150  022767  177777  164256  CMP     #177777,$SP     ;CHECK THE STACK POINTER (R2)
03900    3040  014156  001402                  BEQ     .+6             ;BRANCH IF OK
04000    3041  014160  104000                  HLT                     ;STACK POINTER (R2) NOT EQUAL TO #177777
04100    3042  014162  000507                  507                     ;THE ERROR NUMBER IS 507
04200    3043
04300    3044  014164  022767  014106  164244  CMP     #RTA65, ANS1    ;CHECK FIS TRAP RETURN ADDRESS
04400    3045  014172  001402                  BEQ     .+6             ;BRANCH IF OK
04500    3046  014174  104001                  HLT+1                   ;FIS TRAP AT WRONG ADDRESS
04600    3047  014176  000510                  510                     ;THE ERROR NUMBER IS 510
04700    3048
04800    3049  014200  022767  000151  164232  CMP     #151,   ANS2    ;CHECK PS BEFORE FIS TRAP
04900    3050  014206  001402                  BEQ     .+6             ;BRANCH IF OK
05000    3051  014210  104002                  HLT+2                   ;PS AT FIS TRAP TIME NOT 151
05100    3052  014212  000511                  511                     ;THE ERROR NUMBER IS 511
05200    3053
05300    3054  014214  022767  070707  164220  CMP     #070707,ANS3    ;CHECK DATA FROM THE STACK
05400    3055  014222  001402                  BEQ     .+6             ;BRANCH IF OK
05500    3056  014224  104004                  HLT+4                   ;DATA ON STACK (070707) CHANGED
05600    3057  014226  000512                  512                     ;THE ERROR NUMBER IS 512
05700    3058
05800    3059  014230  022767  016161  164206  CMP     #016161,ANS4    ;CHECK DATA FROM STACK
05900    3060  014236  001402                  BEQ     .+6             ;BRANCH IF OK
```

```
00100   CVKACC  MACY11 30A(1052)  21-AUG-78  15:28  PAGE 74                                                    SEQ 0082
00200   CVKACC.P11  16-AUG-78 08:41          ADDRESS ERROR TEST
00300
00400     3061  014240  104004                                  HLT+4                   ;DATA ON STACK (016161) CHANGED
00500     3062  014242  000513                                  513                     ;THE ERROR NUMBER IS 513
00600     3063
00700     3064  014244  022767  146314  164174                 CMP     #146314,ANS5    ;CHECK DATA FROM STACK
00800     3065  014252  001402                                  BEQ     .+6             ;BRANCH IF OK
00900     3066  014254  104006                                  HLT+6                   ;DATA ON STACK (146314) CHANGED
01000     3067  014256  000514                                  514                     ;THE ERROR NUMBER IS 514
01100     3068
01200     3069  014260  022767  143434  164162                 CMP     #143434,ANS6    ;CHECK DATA FROM STACK
01300     3070  014266  001402                                  BEQ     .+6             ;BRANCH IF OK
01400     3071  014270  104006                                  HLT+6                   ;DATA ON STACK (143434) CHANGED
01500     3072  014272  000515                                  515                     ;THE ERROR NUMBER IS 515
01600     3073  014274  122767  000065  164102          END65:  CMPB    #65,    $TESTN  ;CHECK THE TEST NUMBER
01700     3074  014302  001402                                  BEQ     .+6             ;BRANCH IF OK
01800     3075  014304  104000                                  HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
01900     3076  014306  000516                                  516                     ;THE ERROR NUMBER IS 516
```

```
00100
00200
00300
00400    3077
00500    3078
00600    3079
00700    3080
00800    3081
00900    3082
01000    3083  014310  104400                    SCOPE
01100    3084  014312  012737  014402  000004    TST66:  MOV     #ISR66, @#4     ;SET UP ADDRESS TRAP VECTOR
01200    3085  014320  012737  000340  000006            MOV     #340,   @#6
01300    3086  014326  004567  001654                    JSR     R5,     PUSHR   ;PUSH 4 WORDS ONTO R5 STACK, SET PRIORITY
01400    3087  014332  065432  123456                    .WORD   065432,123456   ;SECOND OPERAND ON TOP
01500    3088  014336  037654  032107                    .WORD   037654,032107   ;FIRST OPERAND ON BOTTOM
01600    3089  014342  000202                            .WORD   202             ;PROCESSOR PRIORITY LEVEL
01700    3090  014344  016456  000340                    .WORD   TRAPER, 340     ;FIS TRAP VECTOR
01800    3091  014350                                    MTPS    #202            ;SET PROCESSOR STATUS
01900    3092  014350  106427                            .WORD   106400!..C
02000    3093  014354  012705  160000                    MOV     #160000,R5      ;SET UP R5 AS STACK POINTER
02100    3094
02200    3095  014360  000240                            NOP
02300    3096  014362  075025                            FMUL    R5              ;FLOATING MULTIPLY ON THE R5 STACK
02400    3097
02500    3098  014364                            RTA66:  MFPS    $PSW            ;SAVE THE PSW
02600    3099  014364  106767                            .WORD   106700!..C
02700    3100  014370  010567  164040                    MOV     R5,     $SP     ;SAVE STACK POINTER (R5)
02800    3101  014374  104000                            HLT                     ;FIS TRAP DIDN'T OCCURE!
02900    3102  014376  000517                            517                     ;THE ERROR NUMBER IS 517
03000    3103  014400  000434                            BR      END66
03100    3104
03200    3105  014402  004767  001662            ISR66:  JSR     PC,     POPER   ;POP ALL DATA OFF THE STACKS
03300    3106  014406  010567  164022                    MOV     R5,     $SP     ;SAVE STACK POINTER (R5)
03400    3107  014412  122767  000340  164012            CMPB    #340,   $PSW    ;CHECK PS AFTER ADR. ERR. TRAP
03500    3108  014420  001402                            BEQ     .+6             ;BRANCH IF OK
03600    3109  014422  104000                            HLT                     ;PS AFTER TRAP NOT EQUAL TO 340
03700    3110  014424  000520                            520                     ;THE ERROR NUMBER IS 520
03800    3111
03900    3112  014426  022767  160000  164000            CMP     #160000,$SP     ;CHECK THE STACK POINTER (R5)
04000    3113  014434  001402                            BEQ     .+6             ;BRANCH IF OK
04100    3114  014436  104000                            HLT                     ;STACK POINTER (R5) NOT EQUAL TO #160000
04200    3115  014440  000521                            521                     ;THE ERROR NUMBER IS 521
04300    3116
04400    3117  014442  022767  014364  163766            CMP     #RTA66, ANS1    ;CHECK FIS TRAP RETURN ADDRESS
04500    3118  014450  001402                            BEQ     .+6             ;BRANCH IF OK
04600    3119  014452  104001                            HLT+1                   ;FIS TRAP AT WRONG ADDRESS
04700    3120  014454  000522                            522                     ;THE ERROR NUMBER IS 522
04800    3121
04900    3122  014456  022767  000210  163754            CMP     #210,   ANS2    ;CHECK PS BEFORE FIS TRAP
05000    3123  014464  001402                            BEQ     .+6             ;BRANCH IF OK
05100    3124  014466  104002                            HLT+2                   ;PS AT FIS TRAP TIME NOT 210
05200    3125  014470  000523                            523                     ;THE ERROR NUMBER IS 523
05300    3126
05400    3127  014472  122767  000066  163704    END66:  CMPB    #66.    $TESTN  ;CHECK THE TEST NUMBER
05500    3128  014500  001402                            BEQ     .+6             ;BRANCH IF OK
05600    3129  014502  104000                            HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
05700    3130  014504  000524                            524                     ;THE ERROR NUMBER IS 524
```

```
00100   CVKACC  MACY11 30A(1052)  21-AUG-78  15:28  PAGE 76
00200   CVKACC.P11    16-AUG-78 08:41          ADDRESS ERROR TEST                                                    SEQ 0084
00300
00400     3131  014506  012737  000006  000004  52200          MOV     #6,     @#4     ;RESTORE TIME-OUT VECTOR
00500     3132  014514  005037  000006          52300          CLR     @#6
00600     3133  014520  012767  000003  164010  52700          MOV     #3,     TIMES   ;REDUCE NUMBER OF ITTERATIONS
00700     3134
00800     3135                                                ;*************************************************************************
00900     3136                                                ;TEST 67:       TEST THAT FIS ABORTS PROPERLY WHEN INTERUPTED
01000     3137                                                ;       035700,143235 + 000177,134543 = 035700,143235
01100     3138                                                ;       PS = .PS,       STACK POINTER = R1
01200     3139                                                ;*************************************************************************
01300     3140
01400     3141  014526  104400                                        SCOPE
01500     3142  014530  132737  000040  000421          TST67:  BITB    #40,@#$ENVM
01600     3143  014536  001177                                  BNE     END67+2         ;EXIT THIS TEST IF BIT 5 OF $ENVM IS HIGH
01700     3144  014540  013704  000546                          MOV     @#TTYOUT,R4
01800     3145  014544  012724  014646                          MOV     #ISR67, (R4)+   ;SET UP TELEPRINTER INTERUPT VECTOR
01900     3146  014550  012714  000340                          MOV     #340,   (R4)
02000     3147  014554  032737  004000  000422                  BIT     #SW11,@#$SWREG  ;TEST FOR ITERATIONS
02100     3148  014562  001005                                  BNE     1$              ;BRANCH TO AVOID HANG UP
02200     3149  014564  000004  000473                          TYPE,   RETURN+1        ;RETURN+1 CAN BE REPLACED WITH THE ADDRESS O
02300     3150                                                                          ;TO TYPE CARRIAGE RETURN, LINE FEED
02400     3151  014570  012767  014576  163672                  MOV     #.+6,   LAD$    ;RESET LOOP ADDRESS
02500     3152  014576  004567  001404          1$:     JSR     R5,     PUSHR   ;PUSH 4 WORDS ONTO R1 STACK, SET PRIORITY
02600     3153  014602  000177  134543                          .WORD   000177,134543   ;SECOND OPERAND ON TOP
02700     3154  014606  035700  143235                          .WORD   035700,143235   ;FIRST OPERAND ON BOTTOM
02800     3155  014612  000143                                  .WORD   143             ;PROCESSOR PRIORITY LEVEL
02900     3156  014614  016456  000340                          .WORD   TRAPER, 340     ;FIS TRAP VECTOR
03000     3157  014620  012701  000510                          MOV     #STACK0,R1      ;SET UP STACK POINTER
03100     3158  014624  012767  000060  163702                  MOV     #60,TEMP        ;INITIALIZE COUNTER FOR FIS INTERRUPTS
03200     3159  014632  112777  000100  163710                  MOVB    #100,   @$TPS   ;SET TTY INTERUPT ENABLE
03300     3160
03400     3161  014640  075001                  RTA67:  FADD    R1                      ;FLOATING ADD ON THE STACK
03500     3162  014642  024141                          CMP     -(R1),  -(R1)           ;RESET THE STACK POINTER FOR NEXT PASS
03600     3163  014644  000775                          BR      RTA67                   ;REPEAT UNTIL INTERUPTED
03700     3164
03800     3165  014646  105077  163676          ISR67:  CLRB    @$TPS                   ;CLEAR THE INTERUPT ENABLE
03900     3166  014652  022716  014640                  CMP     #RTA67, (SP)            ;CHECK IF INTERUPT AT FIS INSTR.
04000     3167  014656  001424                          BEQ     3$                      ;BRANCH IF IT DID
04100     3168  014660  022766  014640  000004          CMP     #RTA67, 4(SP)           ;CHECK FOR INTERUPT WITH T-BIT SET
04200     3169  014666  001423                          BEQ     4$                      ;BRANCH IF IT DID
04300     3170  014670  005337  000534                  DEC     @#TEMP                  ;DID INTERRUPTS OCCUR TEMP # OF TIMES
04400     3171  014674  001517                          BEQ     END67                   ;IF YES THEN SKIP TO END OF THIS TEST
04500     3172  014676  112777  000015  163646  1$:     MOVB    #15,    @$TPB           ;CONTINUE TO TYPE ''CR''
04600     3173  014704  105777  163640          2$:     TSTB    @$TPS                   ;LOOP HERE UNTILL DONE BIT COMES ON
04700     3174  014710  100375                          BPL     2$
04800     3175  014712  112777  000015  163632          MOVB    #15,    @$TPB           ;TYPE ANOTHER ''CR''
04900     3176  014720  012777  000100  163622          MOV     #100,   @$TPS           ;SET TTY INTERUPT ENABLE
05000     3177  014726  000002                          RTI
05100     3178
05200     3179  014730  004767  001334          3$.     JSR     PC,     POPER   ;SAVE ALL THE STUFF ON THE STACK
05300     3180  014734  000403                          BR      5$
05400     3181
05500     3182  014736  022626                  4$:     CMP     (SP)+,  (SP)+   ;RESET THE STACK TO IGNORE THE TRACE TRAP
05600     3183  014740  004767  001330                  JSR     PC,     POPER1  ;POP ALL THE STUFF OFF THE STACK
05700     3184  014744  005746                  5$:     TST     -(SP)           ;SAVE PSW FOR FUTURE RTI
05800     3185  014746  012746  014640                  MOV     #RTA67,-(SP)    ;PLACE THE RTI ADDRESS BACK IN SP
05900     3186  014752  022706  000574                  CMP     #BEGIN-4,SP     ;CHECK THE STACK POINTER
```

```
00300
00400    3187  014756  001407                          BEQ     6$              ;BRANCH IF OK
00500    3188  014760  010667  163450                  MOV     SP,     $SP     ;SAVE FOR TYPING
00600    3189  014764  012706  000574                  MOV     #BEGIN-4,SP     ;RESTORE THE STACK POINTER
00700    3190  014770  104000                          HLT                     ;STACK POINTER FOULED UP
00800    3191  014772  000525                          525                     ;THE ERROR NUMBER IS 525
00900    3192  014774  000457                          BR      END67           ;SKIP REST OF TEST
01000    3193
01100    3194  014776  010167  163432          6$:     MOV     R1,     $SP     ;SAVE STACK POINTER
01200    3195  015002  122767  000344  163422          CMPB    #344,   $PSW    ;CHECK PS AFTER INTERUPT
01300    3196  015010  001402                          BEQ     .+6             ;BRANCH IF OK
01400    3197  015012  104000                          HLT                     ;PS AFTER INTERUPT NOT EQUAL TO LVLA
01500    3198  015014  000526                          526                     ;THE ERROR NUMBER IS 526
01600    3199
01700    3200  015016  022767  000510  163410          CMP     #STACK0,$SP     ;CHECK THE STACK POINTER (R1)
01800    3201  015024  001402                          BEQ     .+6             ;BRANCH IF OK
01900    3202  015026  104000                          HLT                     ;STACK POINTER (R1) NOT EQUAL TO #STACK0
02000    3203  015030  000527                          527                     ;THE ERROR NUMBER IS 527
02100    3204
02200    3205  015032  022767  014640  163376          CMP     #RTA67, ANS1    ;CHECK FIS TRAP RETURN ADDRESS
02300    3206  015040  001402                          BEQ     .+6             ;BRANCH IF OK
02400    3207  015042  104001                          HLT+1                   ;FIS TRAP AT WRONG ADDRESS
02500    3208  015044  000530                          530                     ;THE ERROR NUMBER IS 530
02600    3209
02700    3210
02800    3211  015046  022767  000177  163366          CMP     #000177,ANS3    ;CHECK DATA FROM THE STACK
02900    3212  015054  001402                          BEQ     .+6             ;BRANCH IF OK
03000    3213  015056  104004                          HLT+4                   ;DATA ON STACK (000177) CHANGED
03100    3214  015060  000531                          531                     ;THE ERROR NUMBER IS 531
03200    3215
03300    3216  015062  022767  134543  163354          CMP     #134543,ANS4    ;CHECK DATA FROM STACK
03400    3217  015070  001402                          BEQ     .+6             ;BRANCH IF OK
03500    3218  015072  104004                          HLT+4                   ;DATA ON STACK (134543) CHANGED
03600    3219  015074  000532                          532                     ;THE ERROR NUMBER IS 532
03700    3220
03800    3221  015076  022767  035700  163342          CMP     #035700,ANS5    ;CHECK DATA FROM STACK
03900    3222  015104  001402                          BEQ     .+6             ;BRANCH IF OK
04000    3223  015106  104006                          HLT+6                   ;DATA ON STACK (035700) CHANGED
04100    3224  015110  000533                          533                     ;THE ERROR NUMBER IS 533
04200    3225
04300    3226  015112  022767  143235  163330          CMP     #143235,ANS6    ;CHECK DATA FROM STACK
04400    3227  015120  001402                          BEQ     .+6             ;BRANCH IF OK
04500    3228  015122  104006                          HLT+6                   ;DATA ON STACK (143235) CHANGED
04600    3229  015124  000534                          534                     ;THE ERROR NUMBER IS 534
04700    3230
04800    3231  015126  005367  163402                  DEC     TEMP            ;STAY IN THE LOOP FOR 30 TIMES
04900    3232  015132  001261                          BNE     1$
05000    3233
05100    3234  015134  022626                  END67:  CMP     (SP)+,  (SP)+   ;RESTORE STACK POINTER TO 500
05200    3235  015136  122767  000067  163240          CMPB    #67,    $TESTN  ;CHECK THE TEST NUMBER
05300    3236  015144  001402                          BEQ     .+6             ;BRANCH IF OK
05400    3237  015146  104000                          HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
05500    3238  015150  000535                          535                     ;THE ERROR NUMBER IS 535
05600    3239  015152                                  MTPS    #340
05700    3240  015152  106427                          .WORD   106400...(
05800    3241
```

```
00100
00200
00300
00400    3242
00500    3243                    ;*****************************************************************************
00600    3244                    ;TEST 70:       TEST THAT FIS ABORTS PROPERLY WHEN INTERUPTED
00700    3245                    ;       107070,070707 * 040200,000000 = 107070,070707
00800    3246                    ;       PS = .PS,       STACK POINTER = R0
00900    3247                    ;*****************************************************************************
01000    3248
01100    3249  015156  104400                      SCOPE
01200    3250  015160  132737  000040  000421 TST70: BITB    #40,@#$ENVM
01300    3251  015166  001176                      BNE     END70+2         ;EXIT THIS TEST IF BIT 5 OF $ENVM IS HIGH
01400    3252  015170  013704  000546              MOV     @#TTYOUT,R4
01500    3253  015174  012724  015276              MOV     #ISR70, (R4)+   ;SET UP TELEPRINTER INTERUPT VECTOR
01600    3254  015200  012714  000340              MOV     #340,   (R4)
01700    3255  015204  032737  004000  000422      BIT     #SW11,@#$SWREG  ;TEST FOR ITERATIONS
01800    3256  015212  001005                      BNE     1$              ;BRANCH TO AVOID HANG UP
01900    3257  015214  000004  000473              TYPE,   RETURN+1        ;RETURN+1 CAN BE REPLACED WITH THE ADDRESS O
02000    3258                                                            ;TO TYPE CARRIAGE RETURN, LINE FEED
02100    3259  015220  012767  015226  163242      MOV     #.+6,   LAD$    ;RESET LOOP ADDRESS
02200    3260  015226  004567  000754      1$:     JSR     R5,     PUSHR   ;PUSH 4 WORDS ONTO R0 STACK, SET PRIORITY
02300    3261  015232  040200  000000              .WORD   040200,000000   ;SECOND OPERAND ON TOP
02400    3262  015236  107070  070707              .WORD   107070,070707   ;FIRST OPERAND ON BOTTOM
02500    3263  015242  000100                      .WORD   100             ;PROCESSOR PRIORITY LEVEL
02600    3264  015244  016456  000340              .WORD   TRAPER, 340     ;FIS TRAP VECTOR
02700    3265  015250  012700  000510              MOV     #STACK0,R0      ;SET UP STACK POINTER
02800    3266  015254  012767  000060  163252      MOV     #60,TEMP        ;INITIALIZE COUNTER FOR FIS INTERRUPTS
02900    3267  015262  112777  000100  163260      MOVB    #100,   @$TPS   ;SET TTY INTERUPT ENABLE
03000    3268
03100    3269  015270  075020              RTA70:  FMUL    R0              ;FLOATING MULTIPLY ON THE STACK
03200    3270  015272  024040                      CMP     -(R0),  -(R0)   ;RESET THE STACK POINTER FOR NEXT PASS
03300    3271  015274  000775                      BR      RTA70           ;REPEAT UNTIL INTERUPTED
03400    3272
03500    3273  015276  105077  163246      ISR70:  CLRB    @$TPS           ;CLEAR THE INTERUPT ENABLE
03600    3274  015302  022716  015270              CMP     #RTA70, (SP)    ;CHECK IF INTERUPT AT FIS INSTR.
03700    3275  015306  001424                      BEQ     3$              ;BRANCH IF IT DID
03800    3276  015310  022766  015270  000004      CMP     #RTA70, 4(SP)   ;CHECK FOR INTERUPT WITH T-BIT SET
03900    3277  015316  001423                      BEQ     4$              ;BRANCH IF IT DID
04000    3278  015320  005337  000534              DEC     @#TEMP          ;DID INTERRUPTS OCCUR TEMP # OF TIMES
04100    3279  015324  001516                      BEQ     END70           ;IF YES THEN SKIP TO END OF THIS TEST
04200    3280  015326  112777  000015  163216  1$: MOVB    #15,    @$TPB   ;CONTINUE TO TYPE "CR"
04300    3281  015334  105777  163210      2$:     TSTB    @$TPS           ;LOOP HERE UNTILL DONE BIT COMES ON
04400    3282  015340  100375                      BPL     2$
04500    3283  015342  112777  000015  163202      MOVB    #15,    @$TPB   ;TYPE ANOTHER "CR"
04600    3284  015350  012777  000100  163172      MOV     #100,   @$TPS   ;SET TTY INTERUPT ENABLE
04700    3285  015356  000002                      RTI
04800    3286
04900    3287  015360  004767  000704      3$:     JSR     PC,     POPER   ;SAVE ALL THE STUFF ON THE STACK
05000    3288  015364  000403                      BR      5$
05100    3289
05200    3290  015366  022626              4$.     CMP     (SP)+,  (SP)+   ;RESET THE STACK TO IGNORE THE TRACE TRAP
05300    3291  015370  004767  000700              JSR     PC,     POPER1  ;POP ALL THE STUFF OFF THE STACK
05400    3292  015374  005746              5$:     TST     -(SP)           ;SAVE PSW FOR FUTURE RTI
05500    3293  015376  012746  015270              MOV     #RTA70,-(SP)    ;PLACE THE RTI ADDRESS BACK IN SP
05600    3294  015402  022706  000574              CMP     #BEGIN-4,SP     ;CHECK THE STACK POINTER
05700    3295  015406  001407                      BEQ     6$              ;BRANCH IF OK
05800    3296  015410  010667  163020              MOV     SP,     $SP     ;SAVE FOR TYPING
05900    3297  015414  012706  000574              MOV     #BEGIN-4,SP     ;RESTORE THE STACK POINTER
```

```
00100
00200
00300
00400    3298  015420  104000                          HLT                     ;STACK POINTER FOULED UP
00500    3299  015422  000536                          536                     ;THE ERROR NUMBER IS 536
00600    3300  015424  000456                          BR      END70           ;SKIP REST OF TEST
00700    3301
00800    3302  015426  010067  163002            6$:   MOV     R0,     $SP     ;SAVE STACK POINTER
00900    3303  015432  122767  000344  162772          CMPB    #344,   $PSW    ;CHECK PS AFTER INTERUPT
01000    3304  015440  001402                          BEQ     .+6             ;BRANCH IF OK
01100    3305  015442  104000                          HLT                     ;PS AFTER INTERUPT NOT EQUAL TO LVLA
01200    3306  015444  000537                          537                     ;THE ERROR NUMBER IS 537
01300    3307
01400    3308  015446  022767  000510  162760          CMP     #STACK0,$SP     ;CHECK THE STACK POINTER (R0)
01500    3309  015454  001402                          BEQ     .+6             ;BRANCH IF OK
01600    3310  015456  104000                          HLT                     ;STACK POINTER (R0) NOT EQUAL TO #STACK0
01700    3311  015460  000540                          540                     ;THE ERROR NUMBER IS 540
01800    3312
01900    3313  015462  022767  015270  162746          CMP     #RTA70, ANS1    ;CHECK FIS TRAP RETURN ADDRESS
02000    3314  015470  001402                          BEQ     .+6             ;BRANCH IF OK
02100    3315  015472  104001                          HLT+1                   ;FIS TRAP AT WRONG ADDRESS
02200    3316  015474  000541                          541                     ;THE ERROR NUMBER IS 541
02300    3317
02400    3318
02500    3319  015476  022767  040200  162736          CMP     #040200,ANS3    ;CHECK DATA FROM THE STACK
02600    3320  015504  001402                          BEQ     .+6             ;BRANCH IF OK
02700    3321  015506  104004                          HLT+4                   ;DATA ON STACK (040200) CHANGED
02800    3322  015510  000542                          542                     ;THE ERROR NUMBER IS 542
02900    3323
03000    3324  015512  005767  162726                  TST     ANS4            ;CHECK DATA FROM STACK
03100    3325  015516  001402                          BEQ     .+6             ;BRANCH IF OK
03200    3326  015520  104004                          HLT+4                   ;DATA ON STACK (000000) CHANGED
03300    3327  015522  000543                          543                     ;THE ERROR NUMBER IS 543
03400    3328
03500    3329  015524  022767  107070  162714          CMP     #107070,ANS5    ;CHECK DATA FROM STACK
03600    3330  015532  001402                          BEQ     .+6             ;BRANCH IF OK
03700    3331  015534  104006                          HLT+6                   ;DATA ON STACK (107070) CHANGED
03800    3332  015536  000544                          544                     ;THE ERROR NUMBER IS 544
03900    3333
04000    3334  015540  022767  070707  162702          CMP     #070707,ANS6    ;CHECK DATA FROM STACK
04100    3335  015546  001402                          BEQ     .+6             ;BRANCH IF OK
04200    3336  015550  104006                          HLT+6                   ;DATA ON STACK (070707) CHANGED
04300    3337  015552  000545                          545                     ;THE ERROR NUMBER IS 545
04400    3338
04500    3339  015554  005367  162754                  DEC     TEMP            ;STAY IN THE LOOP FOR 30 TIMES
04600    3340  015560  001262                          BNE     1$
04700    3341
04800    3342  015562  022626                  END70:  CMP     (SP)+,  (SP)+   ;RESTORE STACK POINTER TO 500
04900    3343  015564  122767  000070  162612          CMPB    #70.,   $TESTN  ;CHECK THE TEST NUMBER
05000    3344  015572  001402                          BEQ     .+6             ;BRANCH IF OK
05100    3345  015574  104000                          HLT                     ;WRONG TEST!  PC MUST HAVE FOULED UP.
05200    3346  015576  000546                          546                     ;THE ERROR NUMBER IS 546
05300    3347  015600                                  MTPS    #340
05400    3348  015600  106427                          .WORD   106400+..(
05500    3349
```

```
00100   CVKACC  MACY11 30A(1052)  21-AUG-78  15:28  PAGE 80
00200   CVKACC.P11    16-AUG-78 08:41          INTERUPT ABORT TEST SECTION                                    SEQ 0088
00300
00400     3350  015604  012767  000377  162724  53000          MOV     #377,   TIMES   ;SET NUMBER OF ITERATIONS TO 377
00500     3351  015612  010477  162730          53100          MOV     R4,     @TTYOUT ;RESTORE TTY INTERUPT VECTOR
00600     3352  015616  005014                  53200          CLR     (R4)
00700     3353                                  53300
```

```
00100
00200
00300
00400    3354                                ;*********************************************************************
00500    3355
00600    3356                                .SBTTL  END OF PASS ROUTINE
00700    3357
00800    3358                                ;*INCREMENT THE PASS NUMBER ($PASS)
00900    3359                                ;*TYPE 'END PASS'
01000    3360                                ;*IF THERES A MONITOR GO TO IT
01100    3361                                ;*IF THERE ISN'T JUMP TO BEGIN
01200    3362                                ;*IF IT IS DESIRED TO HAVE A BELL INDICATE THE 'END OF PASS' LOCATION
01300    3363                                ;*$ENDMG CAN BE CHANGED TO 7.
01400    3364
01500    3365  015620                        $EOP:
01600    3366  015620  104400                        SCOPE
01700    3367  015622  005267  162560                INC     $PASS           ;;INCREMENT THE PASS NUMBER
01800    3368  015626  042767  100000  162552      BIC     #100000,$PASS   ;;DON'T ALLOW A NEG. NUMBER
01900    3369  015634  005327                      DEC     (PC)+           ;;LOOP?
02000    3370  015636  000001              $EOPCT: .WORD   1
02100    3371  015640  003015                      BGT     $DOAGN          ;;YES
02200    3372  015642  012737                      MOV     (PC)+,@(PC)+    ;;RESTORE COUNTER
02300    3373  015644  000001              $ENDCT: .WORD   1
02400    3374  015646  015636                      $EOPCT
02500    3375  015650  000004  015700              TYPE    ,$ENDMG         ;;TYPE 'END PASS'
02600    3376  015654                      $GET42:
02700    3377
02800    3378  015654  013700  000042              MOV     @#42,R0         ;;GET MONITOR ADDRESS
02900    3379  015660  001405                      BEQ     $DOAGN          ;;BRANCH IF NO MONITOR
03000    3380  015662  000005                      RESET                   ;;CLEAR THE WORLD
03100    3381  015664  004710              $ENDAD: JSR     PC,(R0)         ;;GO TO MONITOR
03200    3382  015666  000240                      NOP                     ;;SAVE ROOM
03300    3383  015670  000240                      NOP                     ;;FOR
03400    3384  015672  000240                      NOP                     ;;ACT11
03500    3385  015674                      $DOAGN:
03600    3386  015674  000137  000600              JMP     @#BEGIN         ;;RETURN
03700    3387  015700  005015  047105  020104  $ENDMG: .ASCII  <15><12>/END PASS/
03800    3388  015706  040520  051523
03900    3389  015712     377     377     000  $ENULL: .BYTE   -1,-1,0         ;;NULL CHARACTER STRING
04000    3390            015716                      .EVEN
04100    3391                                                53600
04200    3392  015644  000001                    ENDCT:  1                   54100
```

```
00100
00200
00300
00400    3393                                   ;*****************************************************************
00500    3394
00600    3395                                   .SBTTL  SCOPE ROUTINE
00700    3396
00800    3397  015716  032737  000400  000422  SCOPE$: BIT     #SW08,@#$SWREG  ;KILL LDUB OR LOOP ON SPEC. TEST
00900    3398  015724  001404                          BEQ     1$
01000    3399  015726  123767  000422  162450          CMPB    @#$SWREG,$TESTN ;ON RIGHT TEST?   *SW7-0*
01100    3400  015734  001431                          BEQ     OVER$
01200    3401  015736  032737  040000  000422  1$:     BIT     #SW14,@#$SWREG  ;LOOP ON TEST
01300    3402  015744  001023                          BNE     KIT$
01400    3403  015746  032737  004000  000422          BIT     #SW11,@#$SWREG  ;KILL ITERATIONS
01500    3404  015754  001412                          BEQ     SVLAD$
01600    3405  015756  105767  162521                  TSTB    $ICNT
01700    3406  015762  001404                          BEQ     2$              ;BRANCH IF FIRST
01800    3407  015764  126767  162546  162511          CMPB    TIMES,$ICNT     ;DONE?
01900    3408  015772  001010                          BNE     KIT$            ;BRANCH IF NOT
02000    3409  015774  112767  000001  162501  2$:     MOVB    #1,$ICNT        ;FIRST ITERATION
02100    3410  016002  105267  162376          SVLAD$: INCB    $TESTN          ;COUNT TEST NUMBERS
02200    3411  016006  011667  162456                  MOV     (6),LAD$        ;SAVE LOOP ADDRESS
02300    3412  016012  000002                          RTI                     ;RETURN
02400    3413
02500    3414  016014  105267  162463  KIT$:   INCB    $ICNT
02600    3415  016020  005767  162444  OVER$:  TST     LAD$            ;FIRST ONE?
02700    3416  016024  001766                          BEQ     SVLAD$
02800    3417  016026  016716  162436                  MOV     LAD$,(6)        ;FUDGE RETURN ADDRESS
02900    3418  016032  000002                          RTI                     ;FIXES PS
03000    3419
```

```
00100    CVKACC  MACY11 30A(1052)  21-AUG-78  15:28  PAGE 83                                                    SEQ 0091
00200    CVKACC.P11    16-AUG-78 08:41          PUSH AND POP SUBROUTINES
00300
00400     3420                                                55000
00500     3421                                                55100   ;SUBROUTINE TO PUSH 4 WORDS ONTO THE STACK
00600     3422                                                55200
00700     3423  016034  005726                                55300   PUSHS:  TST     (SP)+           ;POP STACK BY 1
00800     3424  016036  062705  000010                        55400           ADD     #10,    R5      ;POINT TO END OF DATA
00900     3425  016042  014546                                55500           MOV     -(R5),  -(SP)   ;PUSH DATA ONTO THE STACK
01000     3426  016044  014546                                55600           MOV     -(R5),  -(SP)   ;PUSH DATA ONTO THE STACK
01100     3427  016046  014546                                55700           MOV     -(R5),  -(SP)   ;PUSH DATA ONTO THE STACK
01200     3428  016050  014546                                55800           MOV     -(R5),  -(SP)   ;PUSH DATA ONTO THE STACK
01300     3429  016052  062705  000010                        55900           ADD     #10,    R5      ;POINT TO END OF DATA
01400     3430  016056                                        56300           MTPS    (R5)+           ;SET THE PROCESSOR STATUS
01500     3431  016056  106425                                                .WORD   106400!..C
01600     3432  016060  005205                                56700           INC     R5
01700     3433  016062  012577  162376                        56800           MOV     (R5)+,  @FISVEC ;SET UP FIS ERROR TRAP VECTOR
01800     3434  016066  012577  162374                        56900           MOV     (R5)+,  @FISLVL ;TRAP STATUS
01900     3435  016072  000115                                57000           JMP     (R5)            ;RETURN
02000     3436                                                57100
02100     3437                                                57200
02200     3438                                                57300   ;SUBROUTINE TO POP 2 WORDS OFF THE STACK
02300     3439                                                57400   ;ALSO SAVES THE PROCESSOR STATUS WORD (EXCEPT T BIT)
02400     3440                                                57500
02500     3441  016074                                        57900   POPS:   MFPS    $PSW            ;SAVE PROCESSOR STATUS WORD
02600     3442  016074  106767                                                .WORD   106700!..C
02700     3443  016100  042767  000020  162324                58300           BIC     #20,    $PSW    ;CLEAR T-BIT
02800     3444  016106  012604                                58400           MOV     (SP)+,  R4      ;SAVE RTS ADDRESS
02900     3445  016110  012667  162322                        58500           MOV     (SP)+,  ANS1    ;SAVE THE ANSWER
03000     3446  016114  012667  162320                        58600           MOV     (SP)+,  ANS2    ;
03100     3447  016120  010667  162310                        58700           MOV     SP,     $SP     ;SAVE THE STACK POINTER
03200     3448  016124  000114                                58800           JMP     (R4)            ;RETURN
03300     3449                                                58900
03400     3450                                                59000
03500     3451                                                59100   ;SUBROUTNE TO POP 6 WORDS OFF THE STACK.
03600     3452                                                59200   ;THE FIRST TWO WERE PUT ON BY THE ERROR TRAP,
03700     3453                                                59300   ;THE LAST FOUR WERE THE ORIGINAL INPUT DATA.
03800     3454                                                59400   ;ALSO SAVES THE PS AND STACK POINTER.
03900     3455                                                59500
04000     3456  016126                                        59900   POPES:  MFPS    $PSW            ;SAVE PROCESSOR STATUS WORD
04100     3457  016126  106767                                                .WORD   106700!..C
04200     3458  016132  012604                                60300           MOV     (SP)+,  R4      ;SAVE RTS ADDRESS
04300     3459  016134  012667  162276                        60400           MOV     (SP)+,  ANS1    ;SAVE RTI ADDRESS
04400     3460  016140  011667  162274                        60500           MOV     (SP),   ANS2    ;SAVE RTI STATUS
04500     3461  016144  042767  000020  162266                60600           BIC     #20,    ANS2    ;CLEAR THE T-BIT
04600     3462  016152  012746  016160                        60700           MOV     #1$,    -(SP)
04700     3463  016156  000002                                60800           RTI                     ;RESTORE THE PROCESSOR STATUS
04800     3464  016160  012667  162256                        60900   1$:     MOV     (SP)+,  ANS3    ;SAVE DATA
04900     3465  016164  012667  162254                        61000           MOV     (SP)+,  ANS4    ;
05000     3466  016170  012667  162252                        61100           MOV     (SP)+,  ANS5    ;
05100     3467  016174  012667  162250                        61200           MOV     (SP)+,  ANS6    ;
05200     3468  016200  010667  162230                        61300           MOV     SP,     $SP     ;SAVE SP
05300     3469  016204  000114                                61400           JMP     (R4)            ;RTS
05400     3470                                                61500
05500     3471                                                61600   ;SUBROUTINE TO PUSH 4 WORDS ONTO THE STACK
05600     3472                                                61700
05700     3473  016206  012704  000510                        61800   PUSHR:  MOV     #STACK0,R4      ;SET R4 TO STACK
05800     3474  016212  012524                                61900           MOV     (R5)+,  (R4)+   ;PUT DATA ON STACK
05900     3475  016214  012524                                62000           MOV     (R5)+,  (R4)+   ;
```

```
00100
00200
00300
00400    3476  016216  012524                          62100          MOV     (R5)+,  (R4)+   ;
00500    3477  016220  012524                          62200          MOV     (R5)+,  (R4)+   ;
00600    3478  016222                                  62600          MTPS    (R5)+           ;SET THE PROCESSOR STATUS
00700    3479  016222  106425                                         .WORD   106400!..C
00800    3480  016224  005205                          63000          INC     R5
00900    3481  016226  012577  162232                  63100          MOV     (R5)+,  @FISVEC ;SET UP FIS ERROR TRAP VECTOR
01000    3482  016232  012577  162230                  63200          MOV     (R5)+,  @FISLVL ;TRAP STATUS
01100    3483  016236  000205                          63300          RTS     R5              ;RETURN
01200    3484                                          63400
01300    3485                                          63500
01400    3486                                          63600  ;SUBROUTINE TO POP 2 WORDS OFF THE STACK
01500    3487                                          63700  ;ALSO SAVES THE PROCESSOR STATUS WORD (EXCEPT T BIT)
01600    3488                                          63800
01700    3489  016240                                  64200  POPR:   MFPS    $PSW            ;SAVE PROCESSOR STATUS WORD
01800    3490  016240  106767                                         .WORD   106700!..C
01900    3491  016244  042767  000020  162160          64600          BIC     #20,    $PSW    ;CLEAR T-BIT
02000    3492  016252  016767  162236  162156          64700          MOV     STACK4, ANS1    ;SAVE THE ANSWER
02100    3493  016260  016767  162232  162152          64800          MOV     STACK6, ANS2    ;
02200    3494  016266  000207                          64900          RTS     PC
02300    3495                                          65000
02400    3496                                          65100
02500    3497                                          65200  ;SUBROUTNE TO POP 6 WORDS OFF THE STACKS.
02600    3498                                          65300  ;THE TWO OFF THE R6 STACK WERE PUT ON BY THE ERROR TRAP,
02700    3499                                          65400  ;THE FOUR OFF THE SOFTWARE STACK WERE THE ORIGINAL INPUT DATA.
02800    3500                                          65500  ;ALSO SAVES THE PS AND STACK POINTER AFTER THE FIS TRAP.
02900    3501                                          65600
03000    3502  016270                                  66000  POPER:  MFPS    $PSW            ;SAVE PROCESSOR STATUS WORD
03100    3503  016270  106767                                         .WORD   106700!..C
03200    3504  016274  012667  000056                  66400  POPER1: MOV     (SP)+,  SAVRTS  ;SAVE RTS ADDRESS
03300    3505  016300  012667  162132                  66500          MOV     (SP)+,  ANS1    ;SAVE RTI ADDRESS
03400    3506  016304  011667  162130                  66600          MOV     (SP),   ANS2    ;SAVE RTI STATUS
03500    3507  016310  042767  000020  162122          66700          BIC     #20,    ANS2    ;CLEAR THE T-BIT
03600    3508  016316  012746  016324                  66800          MOV     #1$,    -(SP)
03700    3509  016322  000002                          66900          RTI                     ;RESTORE PROCESSOR STATUS
03800    3510  016324  016767  162160  162110          67000  1$:     MOV     STACK0, ANS3    ;SAVE DATA
03900    3511  016332  016767  162154  162104          67100          MOV     STACK2, ANS4    ;
04000    3512  016340  016767  162150  162100          67200          MOV     STACK4, ANS5    ;
04100    3513  016346  016767  162144  162074          67300          MOV     STACK6, ANS6    ;
04200    3514  016354  000137                          67400          JMP     @(PC)+          ;SIMULATED RTS
04300    3515  016356  000000                          67500  SAVRTS: 0
04400    3516                                          67600
04500    3517                                          67700  ;SUBROUTINE TO PUSH 4 WORDS ONTO THE PC STACK
04600    3518                                          67800
04700    3519  016360  012504                          67900  PUSH7:  MOV     (R5)+,  R4      ;SET R4 TO STACK
04800    3520  016362  012524                          68000          MOV     (R5)+,  (R4)+   ;PUT DATA ON STACK
04900    3521  016364  012524                          68100          MOV     (R5)+,  (R4)+   ;
05000    3522  016366  012524                          68200          MOV     (R5)+,  (R4)+   ;
05100    3523  016370  012524                          68300          MOV     (R5)+,  (R4)+   ;
05200    3524  016372                                  68700          MTPS    (R5)+           ;SET THE PROCESSOR STATUS
05300    3525  016372  106425                                         .WORD   106400!..C
05400    3526  016374  005205                          69100          INC     R5
05500    3527  016376  012577  162062                  69200          MOV     (R5)+,  @FISVEC ;SET UP FIS ERROR TRAP VECTOR
05600    3528  016402  012577  162060                  69300          MOV     (R5)+,  @FISLVL ;TRAP STATUS
05700    3529  016406  000205                          69400          RTS     R5              ;RETURN
05800    3530                                          69500
05900    3531                                          69600  ;SUBROUTINE TO POP 4 WORDS OFF THE PC ''STACK''
```

```
00300
00400    3532                                          69700   ;ALSO SAVES THE PROCESSOR STATUS WORD (EXCEPT T BIT)
00500    3533                                          69800
00600    3534  016410                                  70200   POP7:   MFPS    $PSW            ;SAVE PROCESSOR STATUS WORD
00700    3535  016410  106767                                          .WORD   106700!..C
00800    3536  016414  042767  000020  162010          70600           BIC     #20,    $PSW    ;CLEAR T-BIT
00900    3537  016422  011600                          70700           MOV     (SP),   R0      ;GET RETURN ADDRESS
01000    3538  016424  162700  000014                  70800           SUB     #14,    R0      ;POINT TO TOP OF ''PC STACK''
01100    3539  016430  012067  162002                  70900           MOV     (R0)+,  ANS1    ;SAVE 1ST HALF INPUT DATA
01200    3540  016434  012067  162000                  71000           MOV     (R0)+,  ANS2    ;SAVE 2ND HALF INPUT DATA
01300    3541  016440  010067  161770                  71100           MOV     R0,     $SP     ;SAVE ASSUMED END PC ''STACK POINTER''
01400    3542  016444  012067  161772                  71200           MOV     (R0)+,  ANS3    ;SAVE 1ST HALF OF ANSWER
01500    3543  016450  012067  161770                  71300           MOV     (R0)+,  ANS4    ;SAVE 2ND HALF OF ANSWER
01600    3544  016454  000207                          71400           RTS     PC
01700    3545                                          71500
01800    3546                                          71600   ;ERRONIOUS TRAP SERVICE ROUTINE
01900    3547                                          71700
02000    3548  016456  104000                          71800   TRAPER: HLT                     ;FIS SHOULDN'T HAVE TRAPED
02100    3549  016460  000547                                          547                     ;THE ERROR NUMBER IS 547
02200    3550  016462  000002                          72000           RTI
02300    3551                                          72100
```

```
00300
00400    3552                                   ;*****************************************************************
00500    3553                                   .SBTTL  HLT ROUTINE (ERROR TYPEOUT)
00600    3554  016464  032737  002000  000422   HLT$:   BIT     #SW10,@#$SWREG  ;SHOULD IT RING THE BELL ON ERROR?
00700    3555  016472  001402                           BEQ     1$              ;NO - SKIP
00800    3556  016474  000004  000504                   TYPE    ,$BELL          ;RING BELL
00900    3557  016500  005267  161756           1$:     INC     ERRORS          ;COUNT THE NUMBER OF ERRORS
01000    3558  016504  032737  020000  000422           BIT     #SW13,@#$SWREG  ;SKIP TYPEOUT IF SET
01100    3559  016512  001023                           BNE     2$              ;SKIP TYPEOUTS
01200    3560  016514  000004  000472                   TYPE    ,RETURN
01300    3561  016520  013637  000402                   MOV     @(6)+,@#$FATAL  ;PLACE THE ERROR NUMBER IN LOCATION $FATAL
01400    3562  016524  014667  161700                   MOV     -(6),HLTAD$     ;PUT ADDRESS OF INSTRUCTION ON STACK
01500    3563  016530  162767  000002  161672           SUB     #2,HLTAD$
01600    3564  016536  017605  000000                   MOV     @(6),TTY        ;TYPE @(6) IN OCTAL
01700    3565  016542  004767  000124                   JSR     %7,PRINTR       ;TYPE LEADING ZERO'S
01800    3566  016546  062716  000002                   ADD     #2,(6)          ;ADJUST THE RETURN ADDRESS
01900    3567  016552  000004  000500                   TYPE    ,SPACE+3
02000    3568  016556  004767  000046                   JSR     PC,ERROR$       ;GO TO USER ERROR ROUTINE
02100    3569  016562  105767  161632           2$:     TSTB    $ENV            ;ARE WE RUNNING UNDER APT?
02200    3570  016566  001403                           BEQ     4$              ;IF NOT THEN GO TO 4$
02300    3571  016570  005237  000400                   INC     @#$MSGTY        ;OTHERWISE INFORM APT
02400    3572  016574  000777                           BR      .               ;AND LOOP
02500    3573  016576  005737  000422           4$:     TST     @#$SWREG        ;HALT ON ERROR
02600    3574  016602  100001                           BPL     .+4             ;SKIP IF CONTINUE
02700    3575  016604  000000                           HALT                    ;HALT ON ERROR!
02800    3576  016606  032737  001000  000422           BIT     #SW09,@#$SWREG  ;CHECK FOR INHIBIT LOOP ON ERROR
02900    3577  016614  001001                           BNE     .+4             ;SKIP IF LOOP ON ERROR
03000    3578  016616  000002                           RTI
03100    3579  016620  105067  161657                   CLRB    $ICNT
03200    3580  016624  000167  177164                   JMP     KIT$            ;LOOP ON TEST UNTIL NO ERRORS
03300    3581
03400    3582                                   72300
```

```
00300
00400     3583                                  72500
00500     3584                                        ;*****************************************************************
00600     3585                                  72700
00700     3586                                  72800   .SBTTL  USER ERROR ROUTINE
00800     3587                                  72900
00900     3588  016630  117767  161574  161702 73000   ERROR$: MOVB    @HLTAD$,TYPCNT  ;TYPE COUNT IS LOW BYTE OF HLT
01000     3589  016636  062767  000002  161674 73100           ADD     #2,     TYPCNT  ;TYPE COUNT = X+2
01100     3590  016644  012703  000430         73200           MOV     #HLTAD$,R3      ;TOP OF DATA TO BE TYPED
01200     3591  016650                         73300   ERR1$:
01300     3592  016650  012305                                 MOV     (R3)+,TTY       ;TYPE (R3)+ IN OCTAL
01400     3593  016652  004767  000014                         JSR     %7,PRINTR       ;TYPE LEADING ZERO'S
01500     3594  016656  000004  000501         73400           TYPE,   SPACE+4         ;SPACE
01600     3595  016662  105367  161652         73500           DECB    TYPCNT          ;CHECK FOR DONE
01700     3596  016666  100370                 73600           BPL     ERR1$           ;BRANCH IF NOT DONE
01800     3597  016670  000207                 73700           RTS     PC
01900     3598                                  73800
```

```
00100   CVKACC  MACY11 30A(1052)  21-AUG-78  15:28  PAGE 88
00200   CVKACC.P11    16-AUG-78 08:41          OCTAL WORD & ADDRESS TYPER                                    SEQ 0096
00300
00400     3599  016672  112767  000001  161644          PRINTR: MOVB    #1,.PR          ;SET ZERO FILL SWITCH
00500     3600  016700  000402                                  BR      .+6             ;SKIP
00600     3601  016702  005067  161636                  PRINTS: CLR     .PR             ;SUPRESS LEADING ZERO'S
00700     3602  016706  112767  177772  161631                  MOVB    #-6,.PR+1       ;SET COUNT
00800     3603  016714  010446                                  MOV     R4,-(6)         ;SAVE R4
00900     3604  016716  012704  017020                          MOV     #.PRBUF,R4      ;SET POINTER TO FIRST ASCII CHAR.
01000     3605  016722  105014                                  CLRB    (4)             ;CLEAR FIRST BYTE
01100     3606  016724  000405                                  BR      .PRF            ;ROTATE FIRST BIT
01200     3607  016726  105014                          .PRL:   CLRB    (4)             ;CLEAR BYTE OF CHARACTER
01300     3608  016730  006105                                  ROL     TTY             ;ROTATE BIT INTO C
01400     3609  016732  106114                                  ROLB    (4)             ;PACK IT
01500     3610  016734  006105                                  ROL     TTY             ;ROTATE BIT INTO C
01600     3611  016736  106114                                  ROLB    (4)             ;PACK IT
01700     3612  016740  006105                          .PRF:   ROL     TTY             ;ROTATE BIT INTO C
01800     3613  016742  106114                                  ROLB    (4)             ;PACK IT
01900     3614  016744  105714                                  TSTB    (4)             ;IS IT ZERO?
02000     3615  016746  001402                                  BEQ     .+6             ;SKIP INC
02100     3616  016750  105267  161570                          INCB    .PR             ;SET FILL SWITCH
02200     3617  016754  105767  161564                          TSTB    .PR             ;CHECK FILL SWITCH
02300     3618  016760  001402                                  BEQ     .+6             ;SKIP BITSET
02400     3619  016762  152724  000060                          BISB    #'0,(4)+        ;MAKE INTO ASCII CHAR
02500     3620  016766  105267  161553                          INCB    .PR+1           ;INC COUNT
02600     3621  016772  001355                                  BNE     .PRL            ;REPEAT
02700     3622  016774  022704  017020                          CMP     #.PRBUF,R4      ;EMPTY BUFFER?
02800     3623  017000  001002                                  BNE     .+6             ;SKIP IF NOT
02900     3624  017002  112724  000060                          MOVB    #'0,(4)+        ;LOAD 1 ZERO
03000     3625  017006  105014                                  CLRB    (4)             ;NULL TERMINATOR
03100     3626  017010  000004  017020                          TYPE    ,.PRBUF         ;TYPE IT
03200     3627  017014  012604                                  MOV     (6)+,R4         ;RESTORE R4
03300     3628  017016  000207                                  RTS     PC              ;RETURN
03400     3629
03500     3630  017020  000004                          .PRBUF: .BLKW   4               ;OUTPUT BUFFER
```

```
00300
00400    3631                                   ;*****************************************************************
00500    3632
00600    3633                                   .SBTTL  POWER DOWN AND UP ROUTINES
00700    3634
00800    3635                                   ;POWER DOWN ROUTINE
00900    3636  017030  012737  017152  000024  $PWRDN: MOV     #$ILLUP,@#PWRVEC ;;SET FOR FAST UP
01000    3637  017036  012737  000340  000026         MOV     #340,@#PWRVEC+2 ;;PRIO:7
01100    3638  017044  010046                         MOV     R0,-(SP)        ;;PUSH R0 ON STACK
01200    3639  017046  010146                         MOV     R1,-(SP)        ;;PUSH R1 ON STACK
01300    3640  017050  010246                         MOV     R2,-(SP)        ;;PUSH R2 ON STACK
01400    3641  017052  010346                         MOV     R3,-(SP)        ;;PUSH R3 ON STACK
01500    3642  017054  010446                         MOV     R4,-(SP)        ;;PUSH R4 ON STACK
01600    3643  017056  010546                         MOV     R5,-(SP)        ;;PUSH R5 ON STACK
01700    3644  017060  010667  000072                 MOV     SP,$SAVR6       ;;SAVE SP
01800    3645  017064  012737  017076  000024         MOV     #$PWRUP,@#PWRVEC ;;SET UP VECTOR
01900    3646  017072  000000                         HALT
02000    3647  017074  000776                         BR      .-2             ;;HANG UP
02100    3648
02200    3649                                   ;POWER UP ROUTINE
02300    3650  017076  016706  000054  $PWRUP: MOV     $SAVR6,SP       ;;GET SP
02400    3651  017102  005067  000050         CLR     $SAVR6          ;;WAIT LOOP FOR THE TTY
02500    3652  017106  005267  000044  1$:     INC     $SAVR6          ;;WAIT FOR THE INC
02600    3653  017112  001375                         BNE     1$              ;;OF  WORD
02700    3654  017114  012605                         MOV     (SP)+,R5        ;;POP STACK INTO R5
02800    3655  017116  012604                         MOV     (SP)+,R4        ;;POP STACK INTO R4
02900    3656  017120  012603                         MOV     (SP)+,R3        ;;POP STACK INTO R3
03000    3657  017122  012602                         MOV     (SP)+,R2        ;;POP STACK INTO R2
03100    3658  017124  012601                         MOV     (SP)+,R1        ;;POP STACK INTO R1
03200    3659  017126  012600                         MOV     (SP)+,R0        ;;POP STACK INTO R0
03300    3660  017130  012737  017030  000024         MOV     #$PWRDN,@#PWRVEC ;;SET UP THE POWER DOWN VECTOR
03400    3661  017136  012737  000340  000026         MOV     #340,@#PWRVEC+2 ;;PRIO:7
03500    3662  017144  000004                         TYPE                    ;REPORT THE POWER FAILURE
03600    3663  017146  017160  $PWRMG: .WORD   $POWER          ;;POWER FAIL MESSAGE POINTER
03700    3664  017150  000002                         RTI
03800    3665  017152  000000  $ILLUP: HALT                    ;;THE POWER UP SEQUENCE WAS STARTED
03900    3666  017154  000776                         BR      .-2             ;; BEFORE THE POWER DOWN WAS COMPLETE
04000    3667  017156  000000  $SAVR6: 0                      ;;PUT THE SP HERE
04100    3668  017160  005015  047520  042527  $POWER: .ASCIZ  <15><12>'POWER'
04200    3669  017166  000122
04300    3670                                           .EVEN
```

```
00100
00200
00300
00400    3671                                    74700
00500    3672                                    74800  ;*      TYPE OUT ROUTINE
00600    3673                                    74900  ;*      ----------------
00700    3674                                    75000  ;*
00800    3675                                    75100  ;*
00900    3676                                    75200  ;*      THIS ROUTINE IS USED TO TYPE ASCIZ MESSAGES
01000    3677                                    75300  ;*
01100    3678                                    75400
01200    3679  017170  132737  000040  000421   75500  $TYPE:  BITB    #40,@#$ENVM     ;HAS THE CONSOLE OUTPUTS BEEN SUPPRESSED?
01300    3680  017176  001007                    75600          BNE     3$              ;IF SO THEN RETURN FROM THE SUBROUTINE VIA 3
01400    3681  017200  010046                    75700          MOV     R0,-(SP)        ;OTHERWISE SAVE R0
01500    3682  017202  017600  000002            75800          MOV     @2(SP),R0       ;GET THE ADDRESS OF THE ASSCIZ STRING
01600    3683  017206  112046                    75900  2$:     MOVB    (R0)+,-(SP)     ;PUSH THE CHARACTER TO BE TYPED ONTO STACK
01700    3684  017210  001005                    76000          BNE     4$              ;BRANCH IF IT IS NOT THE TERMINATOR
01800    3685  017212  005726                    76100          TST     (SP)+
01900    3686  017214  012600                    76200          MOV     (SP)+,R0        ;OTHERWISE RESTORE THE STACK AND R0
02000    3687  017216  062716  000002            76300  3$:     ADD     #2,(SP)         ;ADJUST THE RETURN PC
02100    3688  017222  000002                    76400          RTI                     ;AND RETURN
02200    3689                                    76500
02300    3690  017224  105777  161320            76600  4$:     TSTB    @$TPS           ;IS THE PRINTER AVAILABLE?
02400    3691  017230  100375                    76700          BPL     4$              ;IF NOT THEN LOOP HERE
02500    3692  017232  112677  161314            76800          MOVB    (SP)+,@$TPB     ;OUT PUT THE CHARACTER
02600    3693  017236  000763                    76900          BR      2$              ;AND GO BACK
02700    3694  017240  005015  053104  040513   77000  $TTTLE: .ASCIZ  <15><12>/DVKACC - LSI-11 FIS  INSTRUCTION TEST/
02800    3695  017246  041503  026440  046040
02900    3696  017254  044523  030455  020061
03000    3697  017262  044506  020123  044440
03100    3698  017270  051516  051124  041525
03200    3699  017276  044524  047117  052040
03300    3700  017304  051505  000124
03400    3701                                    77100  .EVEN                           ;THAT WAS THE HEADING FOR THE DIAGNOSTIC
03500    3702  017310  012706  000600            77200  NOOOP:  MOV     #BEGIN,SP       ;INITIALIZE STACK POINTER TO 600
03600    3703  017314  132737  000001  000420   77300          BITB    #1,@#$ENV       ;ARE WE UNDER APT
03700    3704  017322  001011                    77400          BNE     22$             ;IF SO THEN DO NOT TYPE HEADING
03800    3705  017324  132737  000040  000421   77450          BITB    #40,@#$ENVM     ;HAVE THE CONSOLE OUTPUTS BEEN SUPPRESSED
03900    3706  017332  001005                    77460          BNE     22$             ;IF SO THEN DO NOT PRINT HEADING
04000    3707  017334  012737  017170  000020   77500          MOV     #$TYPE,@#20     ;SET UP VECTOR 20 TO PRINT HEADING
04100    3708  017342  000004  017240            77600          TYPE    ,$TTTLE         ;TYPE HEADING 'VKACC -LSI-11 ...'
04200    3709  017346  012767  000001  161162   77700  22$:    MOV     #1,TIMES        ; # OF ITERATIONS IN THE FIRST PASS=1
04300    3710  017354  012700  000410            77800          MOV     #$DEVCT,R0      ;PREPARE TO INITIALIZE THE PROGRAM
04400    3711  017360  005040                    77900  2$:     CLR     -(R0)
04500    3712  017362  022700  000400            78000          CMP     #$MAIL,R0
04600    3713  017366  001374                    78100          BNE     2$
04700    3714  017370  000167  160610            78200          JMP     RESTRT          ;START THE PROGRAM
04800    3715          000001                    78300  .END
```

```
A       = 015716          3392#  3393
ABASE   = 000000           100
ACDW1   = 000000           100
ACDW2   = 000000           100
ACPUOP= 000000             100    118
ADDW0   = 000000           100
ADDW1   = 000000           100
ADDW10= 000000             100
ADDW11= 000000             100
ADDW12= 000000             100
ADDW13= 000000             100
ADDW14= 000000             100
ADDW15= 000000             100
ADDW2   = 000000           100
ADDW3   = 000000           100
ADDW4   = 000000           100
ADDW5   = 000000           100
ADDW6   = 000000           100
ADDW7   = 000000           100
ADDW8   = 000000           100
ADDW9   = 000000           100
ADEVCT= 000000             100    109
ADEVM   = 000000           100
AENV    = 000000           100    114
AENVM   = 000000           100    115
AFATAL= 000000             100    106
AMADR1= 000000             100
AMADR2= 000000             100
AMADR3= 000000             100
AMADR4= 000000             100
AMAMS1= 000000             100
AMAMS2= 000000             100
AMAMS3= 000000             100
AMAMS4= 000000             100
AMSGAD= 000000             100    111
AMSGLG= 000000             100    112
AMSGTY= 000000             100    105
AMTYP1= 000000             100
AMTYP2= 000000             100
AMTYP3= 000000             100
AMTYP4= 000000             100
ANS1    000436             156#   157   259   305   351   397   443   489   535   581   627   673    71
                           765    811   855   917   989  1061  1133  1199  1245  1297  1343  1395   144
                          1487   1533  1579  1625  1671  1723  1775  1847  1913  1959  2005  2051   209
                          2143   2187  2241  2303  2375  2441  2487  2533  2579  2623  2685  2757   282
                          2901   2976* 2990  3044  3117  3205  3313  3445* 3459* 3492* 3505* 3539*
ANS2    000440             158#   159   264   310   356   402   448   494   540   586   632   678    72
                           770    816   860   922   994  1066  1138  1204  1250  1302  1348  1400   144
                          1492   1538  1584  1630  1676  1728  1780  1852  1918  1964  2010  2056   210
                          2148   2192  2246  2308  2380  2446  2492  2538  2584  2628  2690  2762   283
                          2906   2977* 2995  3049  3122  3446* 3460* 3461* 3493* 3506* 3507* 3540*
ANS3    000442             160#   161   865   927   999  1071  1143  1785  1857  2197  2251  2313   238
                          2633   2695  2767  2839  2917  3054  3211  3319  3464* 3510* 3542*
ANS4    000444             162#   870   932  1004  1076  1148  1790  1862  2202  2256  2318  2390   263
                          2700   2772  2844  2922  3059  3216  3324  3465* 3511* 3543*
ANS5    000446             163#   937  1009  1081  1153  1795  1867  2323  2395  2705  2777  2849   292
```

```
                                3064    3221    3329    3466*   3512*
ANS6    000450                   164#    942    1014    1086    1158    1800    1872    2328    2400    2710    2782    2854     293
                                3069    3226    3334    3467*   3513*
APASS = 000000                   100     108
APRIOR= 000000                   100
ASWREG= 000000                   100     116
ATESTN= 000000                   100     107
AUNIT = 000000                   100     110
AUSWR = 000000                   100     117
AVECT1= 000000                   100
AVECT2= 000000                   100
BEGIN   000600                   201    208#     431     433     477     479     707     709     753     755     973     977      97
                                1117    1121    1123    1279    1281    1377    1379    1705    1707    1759    1763    1765     208
                                2087    2359    2363    2365    2475    2477    2885    2889    2891    3186    3189    3294     329
                                3386    3702
ENDCT   015644                  3392#
END1    001040                   269#
END10   002254                   591#
END11   002412                   637#
END12   002550                   683#
END13   002702                   712     729#
END14   003036                   758     775#
END15   003176                   821#
END16   003350                   875#
END17   003602                   903     947#
END2    001200                   315#
END20   004032                   974     982    1019#
END21   004270                  1047    1091#
END22   004522                  1118    1126    1163#
END23   004662                  1209#
END24   005016                  1255#
END25   005152                  1290    1307#
END26   005310                  1353#
END27   005444                  1388    1405#
END3    001334                   361#
END30   005604                  1451#
END31   005744                  1497#
END32   006104                  1543#
END33   006242                  1589#
END34   006376                  1635#
END35   006534                  1681#
END36   006670                  1716    1733#
END37   007122                  1760    1768    1805#
END4    001470                   407#
END40   007360                  1833    1877#
END41   007514                  1923#
END42   007654                  1969#
END43   010010                  2015#
END44   010144                  2061#
END45   010300                  2090    2107#
END46   010436                  2153#
END47   010612                  2207#
END5    001622                   436     453#
END50   010766                  2261#
END51   011222                  2289    2333#
END52   011454                  2360    2368    2405#
```

```
END53   011612          2451#
END54   011746          2480    2497#
END55   012104          2543#
END56   012240          2589#
END57   012414          2643#
END6    001754           482     499#
END60   012652          2671    2715#
END61   013110          2743    2787#
END62   013346          2815    2859#
END63   013600          2886    2894    2937#
END64   014016          3000#
END65   014274          3030    3073#
END66   014472          3103    3127#
END67   015134          3143    3171    3192    3234#
END7    002114           545#
END70   015562          3251    3279    3300    3342#
ERRORS  000462           167#   3557*
ERROR$  016630          3568    3588#
ERR1$   016650          3591#   3596
FISLVL  000466           169#   3434*   3482*   3528*
FISVEC  000464           168#   3433*   3481*   3527*
HLT   = 104000            56#    251     256     261     266     271     297     302     307     312     317     343     34
                         353     358     363     389     394     399     404     409     434     440     445     450     45
                         480     486     491     496     501     527     532     537     542     547     573     578     58
                         588     593     619     624     629     634     639     665     670     675     680     685     71
                         716     721     726     731     756     762     767     772     777     803     808     813     81
                         823     852     857     862     867     872     877     901     909     914     919     924     92
                         934     939     944     949     971     980     986     991     996    1001    1006    1011    101
                        1021    1045    1053    1058    1063    1068    1073    1078    1083    1088    1093    1115    112
                        1130    1135    1140    1145    1150    1155    1160    1165    1191    1196    1201    1206    121
                        1237    1242    1247    1252    1257    1288    1294    1299    1304    1309    1335    1340    134
                        1350    1355    1386    1392    1397    1402    1407    1433    1438    1443    1448    1453    147
                        1484    1489    1494    1499    1525    1530    1535    1540    1545    1571    1576    1581    158
                        1591    1617    1622    1627    1632    1637    1663    1668    1673    1678    1683    1714    172
                        1725    1730    1735    1757    1766    1772    1777    1782    1787    1792    1797    1802    180
                        1831    1839    1844    1849    1854    1859    1864    1869    1874    1879    1905    1910    191
                        1920    1925    1951    1956    1961    1966    1971    1997    2002    2007    2012    2017    204
                        2048    2053    2058    2063    2088    2094    2099    2104    2109    2135    2140    2145    215
                        2155    2184    2189    2194    2199    2204    2209    2238    2243    2248    2253    2258    226
                        2287    2295    2300    2305    2310    2315    2320    2325    2330    2335    2357    2366    237
                        2377    2382    2387    2392    2397    2402    2407    2433    2438    2443    2448    2453    247
                        2484    2489    2494    2499    2525    2530    2535    2540    2545    2571    2576    2581    258
                        2591    2620    2625    2630    2635    2640    2645    2669    2677    2682    2687    2692    269
                        2702    2707    2712    2717    2741    2749    2754    2759    2764    2769    2774    2779    278
                        2789    2813    2821    2826    2831    2836    2841    2846    2851    2856    2861    2883    289
                        2898    2903    2914    2919    2924    2929    2934    2939    2981    2986    2992    2997    300
                        3028    3036    3041    3046    3051    3056    3061    3066    3071    3075    3101    3109    311
                        3119    3124    3129    3190    3197    3202    3207    3213    3218    3223    3228    3237    329
                        3305    3310    3315    3321    3326    3331    3336    3345    3548
HLTAD$  000430           150#    151    3562*   3563*   3588    3590
HLT$    016464           214    3554#
ISA20   003710           978     984#
ISA22   004376          1122    1128#
ISA37   006776          1764    1770#
ISA52   011330          2364    2370#
ISA63   013454          2890    2896#
```

```
ISR17   003436    893   905#
ISR20   003664    965   976#
ISR21   004120   1037  1049#
ISR22   004352   1109  1120#
ISR37   006752   1751  1762#
ISR40   007210   1823  1835#
ISR51   011054   2279  2291#
ISR52   011304   2351  2362#
ISR60   012502   2661  2673#
ISR61   012740   2733  2745#
ISR62   013176   2805  2817#
ISR63   013430   2877  2888#
ISR65   014124   3012  3032#
ISR66   014402   3084  3105#
ISR67   014646   3145  3165#
ISR70   015276   3253  3273#
KIT$    016014   3402  3408  3414#  3580
LAD$    000470    170#  226*  3151*  3259*  3411*  3415  3417
N     = 000071     77#   229   275#   321#   367#   413#   459#   505#   551#   597#   643#   689#    73
                  781#  827#  881#  953#  1025#  1097#  1169#  1215#  1261#  1313#  1359#  1411#   145
                 1503#  1549#  1595#  1641#  1687#  1739#  1811#  1883#  1929#  1975#  2021#  2067#   211
                 2159#  2213#  2267#  2339#  2411#  2457#  2503#  2549#  2595#  2649#  2721#  2793#   286
                 2943#  2969  3005#  3077#  3131#  3134  3242#  3350#
NOOOP   017310    200  3702#
OVER$   016020   3400  3415#
POPER   016270    905  1049  1835  2291  2673  2745  2817  3032  3105  3179  3287  3502#
POPER1  016274   3183  3291  3504#
POPES   016126    976  1120  1762  2362  2888  3456#
POPR    016240    247   293   339   385   523   569   615   661   799   899  1043  1187   123
                 1331  1429  1475  1521  1567  1613  1659  1829  1901  1947  1993  2039   213
                 2285  2429  2521  2567  2667  2739  2811  3026  3489#
POPS    016074    430   476   706   752   970  1114  1278  1376  1704  1756  2084  2356   247
                 2882  3441#
POP7    016410    849  2181  2235  2617  3534#
PRINTR  016672   3565  3593  3599#
PRINTS  016702   3601#
PUSHR   016206    237   283   329   375   513   559   605   651   789   889  1033  1177   122
                 1321  1419  1465  1511  1557  1603  1649  1819  1891  1937  1983  2029   212
                 2275  2419  2511  2557  2657  2729  2801  3014  3086  3152  3260  3473#
PUSHS   016034    421   467   697   743   961  1105  1269  1367  1695  1747  2075  2347   246
                 2873  3423#
PUSH7   016360    835  2167  2221  2603  3519#
PWRVEC= 000024     66#  3636*  3637*  3645*  3660*  3661*
RESTRT  000204    201#  3714
RETURN  000472    171#  3149  3257  3560
RTA17   003420    899#   917
RTA20   003646    970#   989
RTA21   004102   1043#  1061
RTA22   004334   1114#  1133
RTA37   006734   1756#  1775
RTA40   007172   1829#  1847
RTA51   011036   2285#  2303
RTA52   011266   2356#  2375
RTA60   012464   2667#  2685
RTA61   012722   2739#  2757
RTA62   013160   2811#  2829
```

```
RTA63   013412          2882#  2901
RTA65   014106          3026#  3044
RTA66   014364          3098#  3117
RTA67   014640          3161#  3163  3166  3168  3185  3205
RTA70   015270          3269#  3271  3274  3276  3293  3313
SAVRTS  016356          3504*  3515#
SAVTPS  000506           177#
SCOPE = 104400            67#   236   282   328   374   420   466   512   558   604   650   696   74
                          788   834   888   960  1032  1104  1176  1222  1268  1320  1366  1418  146
                         1510  1556  1602  1648  1694  1746  1818  1890  1936  1982  2028  2074  212
                         2166  2220  2274  2346  2418  2464  2510  2556  2602  2656  2728  2800  287
                         2952  3011  3083  3141  3249  3366
SCOPE$  015716           216  3397#
SPACE   000475           172#  3567  3594
STACK0  000510           178#   185   242   288   334   380   518   564   610   656   794   894   91
                         1038  1056  1182  1228  1326  1424  1470  1516  1562  1608  1654  1824  184
                         1896  1942  1988  2034  2126  2280  2298  2424  2516  2562  2662  2680  273
                         2752  2806  2824  3157  3200  3265  3308  3473  3510
STACK1= 000511           185#
STACK2  000512           179#  3511
STACK4  000514           180#   254   300   346   392   530   576   622   668   806  1194  1240  133
                         1436  1482  1528  1574  1620  1666  1908  1954  2000  2046  2138  2436  252
                         2574  3492  3512
STACK6  000516           181#  3493  3513
STACK8  000520           182#
STAK10  000532           184#  2953  2984
STK16   003244           836   844#
STK47   010504          2168  2176#
STK50   010660          2222  2230#
STK57   012306          2604  2612#
SVLAD$  016002          3404  3410#  3416
SW08  = 000400            35    75#  3397
SW09  = 001000            36    74#  3576
SW10  = 002000            37    73#  3554
SW11  = 004000            38    72#  3147  3255  3403
SW12  = 010000            39    71#
SW13  = 020000            40    70#  3558
SW14  = 040000            41    69#  3401
SW15  = 100000            42    68#
TEMP    000534           186#  3158*  3170*  3231*  3266*  3278*  3339*
TIMES   000536           187#  3133*  3350*  3407   3709*
TRAPER  016456           241    287    333    379    425    471    517    563    609    655    701    747    79
                         840   1181   1227   1273   1325   1371   1423   1469   1515   1561   1607   1653   169
                        1895   1941   1987   2033   2079   2125   2172   2226   2423   2469   2515   2561   260
                        3018   3090   3156   3264   3548#
TSA13   002640           708    714#
TSA14   002772           754    760#
TSA25   005106          1280   1292#
TSA27   005400          1378   1390#
TSA36   006624          1706   1718#
TSA45   010234          2086   2092#
TSA5    001560           432    438#
TSA54   011702          2476   2482#
TSA6    001712           478    484#
TST1    000722           237#
TST10   002132           559#
```

TST11	002272	605#				
TST12	002430	651#				
TST13	002566	697#				
TST14	002720	743#				
TST15	003054	789#				
TST16	003214	835#				
TST17	003366	889#				
TST2	001056	283#				
TST20	003620	961#				
TST21	004050	1033#				
TST22	004306	1105#				
TST23	004540	1177#				
TST24	004700	1223#				
TST25	005034	1269#				
TST26	005170	1321#				
TST27	005326	1367#				
TST3	001216	329#				
TST30	005462	1419#				
TST31	005622	1465#				
TST32	005762	1511#				
TST33	006122	1557#				
TST34	006260	1603#				
TST35	006414	1649#				
TST36	006552	1695#				
TST37	006706	1747#				
TST4	001352	375#				
TST40	007140	1819#				
TST41	007376	1891#				
TST42	007532	1937#				
TST43	007672	1983#				
TST44	010026	2029#				
TST45	010162	2075#				
TST46	010316	2121#				
TST47	010454	2167#				
TST5	001506	421#				
TST50	010630	2221#				
TST51	011004	2275#				
TST52	011240	2347#				
TST53	011472	2419#				
TST54	011630	2465#				
TST55	011764	2511#				
TST56	012122	2557#				
TST57	012256	2603#				
TST6	001640	467#				
TST60	012432	2657#				
TST61	012670	2729#				
TST62	013126	2801#				
TST63	013364	2873#				
TST64	013616	2953#				
TST65	014034	3012#				
TST66	014312	3084#				
TST67	014530	3142#				
TST7	001772	513#				
TST70	015160	3250#				
TTYOUT	000546	191#	3144	3252	3351*	
TYPCNT	000540	188#	3588*	3589*	3595*	

```
TYPE   = 000004          76#   3149   3257   3375   3556   3560   3567   3594   3626   3662   3708
YESRT    000542         189#    209
$APTHD   000430         136     142#   149
$BELL    000504         176#   3556
$CPUOP   000426         118#
$DEVCT   000410         109#   3710
$DOAGN   015674        3371    3379   3385#
$ENDAD   015664          89    3381#
$ENDCT   015644        3373#   3392
$ENDMG   015700        3375    3387#
$ENULL   015712        3389#
$ENV     000420         114#    219   3569   3703
$ENVM    000421         115#   3142   3250   3679   3705
$EOP     015620        3365#
$EOPCT   015636        3370#   3374
$ETABL   000420         113#
$ETEND   000430         125#    148
$F     = 000550          78#    252    253#   257    258#   262    263#   267    268#   272    273#   298    29
                        303     304#   308    309#   313    314#   318    319#   344    345#   349    350#   35
                        355#    359    360#   364    365#   390    391#   395    396#   400    401#   405    40
                        410     411#   435    436#   441    442#   446    447#   451    452#   456    457#   48
                        482#    487    488#   492    493#   497    498#   502    503#   528    529#   533    53
                        538     539#   543    544#   548    549#   574    575#   579    580#   584    585#   58
                        590#    594    595#   620    621#   625    626#   630    631#   635    636#   640    64
                        666     667#   671    672#   676    677#   681    682#   686    687#   711    712#   71
                        718#    722    723#   727    728#   732    733#   757    758#   763    764#   768    76
                        773     774#   778    779#   804    805#   809    810#   814    815#   819    820#   82
                        825#    853    854#   858    859#   863    864#   868    869#   873    874#   878    87
                        902     903#   910    911#   915    916#   920    921#   925    926#   930    931#   93
                        936#    940    941#   945    946#   950    951#   972    973#   981    982#   987    98
                        992     993#   997    998#  1002   1003#  1007   1008#  1012   1013#  1017   1018#  102
                       1023#   1046   1047#  1054   1055#  1059   1060#  1064   1065#  1069   1070#  1074   107
                       1079    1080#  1084   1085#  1089   1090#  1094   1095#  1116   1117#  1125   1126#  113
                       1132#   1136   1137#  1141   1142#  1146   1147#  1151   1152#  1156   1157#  1161   116
                       1166    1167#  1192   1193#  1197   1198#  1202   1203#  1207   1208#  1212   1213#  123
                       1239#   1243   1244#  1248   1249#  1253   1254#  1258   1259#  1289   1290#  1295   129
                       1300    1301#  1305   1306#  1310   1311#  1336   1337#  1341   1342#  1346   1347#  135
                       1352#   1356   1357#  1387   1388#  1393   1394#  1398   1399#  1403   1404#  1408   140
                       1434    1435#  1439   1440#  1444   1445#  1449   1450#  1454   1455#  1480   1481#  148
                       1486#   1490   1491#  1495   1496#  1500   1501#  1526   1527#  1531   1532#  1536   153
                       1541    1542#  1546   1547#  1572   1573#  1577   1578#  1582   1583#  1587   1588#  159
                       1593#   1618   1619#  1623   1624#  1628   1629#  1633   1634#  1638   1639#  1664   166
                       1669    1670#  1674   1675#  1679   1680#  1684   1685#  1715   1716#  1721   1722#  172
                       1727#   1731   1732#  1736   1737#  1758   1759#  1767   1768#  1773   1774#  1778   177
                       1783    1784#  1788   1789#  1793   1794#  1798   1799#  1803   1804#  1808   1809#  183
                       1833#   1840   1841#  1845   1846#  1850   1851#  1855   1856#  1860   1861#  1865   186
                       1870    1871#  1875   1876#  1880   1881#  1906   1907#  1911   1912#  1916   1917#  192
                       1922#   1926   1927#  1952   1953#  1957   1958#  1962   1963#  1967   1968#  1972   197
                       1998    1999#  2003   2004#  2008   2009#  2013   2014#  2018   2019#  2044   2045#  204
                       2050#   2054   2055#  2059   2060#  2064   2065#  2089   2090#  2095   2096#  2100   210
                       2105    2106#  2110   2111#  2136   2137#  2141   2142#  2146   2147#  2151   2152#  215
                       2157#   2185   2186#  2190   2191#  2195   2196#  2200   2201#  2205   2206#  2210   221
                       2239    2240#  2244   2245#  2249   2250#  2254   2255#  2259   2260#  2264   2265#  228
                       2289#   2296   2297#  2301   2302#  2306   2307#  2311   2312#  2316   2317#  2321   232
                       2326    2327#  2331   2332#  2336   2337#  2358   2359#  2367   2368#  2373   2374#  237
                       2379#   2383   2384#  2388   2389#  2393   2394#  2398   2399#  2403   2404#  2408   240
```

```
                    2434    2435#   2439    2440#   2444    2445#   2449    2450#   2454    2455#   2479    2480#   248
                    2486#   2490    2491#   2495    2496#   2500    2501#   2526    2527#   2531    2532#   2536    253
                    2541    2542#   2546    2547#   2572    2573#   2577    2578#   2582    2583#   2587    2588#   259
                    2593#   2621    2622#   2626    2627#   2631    2632#   2636    2637#   2641    2642#   2646    264
                    2670    2671#   2678    2679#   2683    2684#   2688    2689#   2693    2694#   2698    2699#   270
                    2704#   2708    2709#   2713    2714#   2718    2719#   2742    2743#   2750    2751#   2755    275
                    2760    2761#   2765    2766#   2770    2771#   2775    2776#   2780    2781#   2785    2786#   279
                    2791#   2814    2815#   2822    2823#   2827    2828#   2832    2833#   2837    2838#   2842    284
                    2847    2848#   2852    2853#   2857    2858#   2862    2863#   2884    2885#   2893    2894#   289
                    2900#   2904    2905#   2915    2916#   2920    2921#   2925    2926#   2930    2931#   2935    293
                    2940    2941#   2982    2983#   2987    2988#   2993    2994#   2998    2999#   3003    3004#   302
                    3030#   3037    3038#   3042    3043#   3047    3048#   3052    3053#   3057    3058#   3062    306
                    3067    3068#   3072    3073#   3076    3077#   3102    3103#   3110    3111#   3115    3116#   312
                    3121#   3125    3126#   3130    3131#   3191    3192#   3198    3199#   3203    3204#   3208    320
                    3214    3215#   3219    3220#   3224    3225#   3229    3230#   3238    3239#   3299    3300#   330
                    3307#   3311    3312#   3316    3317#   3322    3323#   3327    3328#   3332    3333#   3337    333
                    3346    3347#   3549    3550#
$FATAL  000402       106#   3561*
$GET42  015654      3376#
$HD   = 000003        14      15
$HIBTS  000430       143#
$ICNT   000503       174#   3405    3407    3409*   3414*   3579*
$ILLUP  017152      3636    3665#
$MAIL   000400       104#    144     148    3712
$MBADR  000432       144#
$MSGAD  000414       111#
$MSGLG  000416       112#
$MSGTY  000400       105#   3571*
$PASS   000406       108#   3367*   3368*   3387
$PASTM  000436       146#
$POWER  017160      3663    3668#
$PSW    000432       152#    153     249     295     341     387     438     484     525     571     617     663      71
                      760     801     850     907     984    1051    1128    1189    1235    1292    1333    1390     143
                     1477    1523    1569    1615    1661    1718    1770    1837    1903    1949    1995    2041     209
                     2133    2182    2236    2293    2370    2431    2482    2523    2569    2618    2675    2747     281
                     2896    2974*   2975*   2979    3034    3099*   3107    3195    3303    3442*   3443*   3457*    349
                     3491*   3503*   3535*   3536*
$PWRDN  017030       211    3636#   3660
$PWRMG  017146      3663#
$PWRUP  017076      3645    3650#
$SAVR6  017156      3644*   3650    3651*   3652*   3667#
$SETUP= 000020        82#   3367
$SP     000434       154#    155     248*    254     294*    300     340*    346     386*    392     524*    530      57
                      576     616*    622     662*    668     800*    806     900*    906*    912    1044*   1050*    105
                     1188*   1194    1234*   1240    1332*   1338    1430*   1436    1476*   1482    1522*   1528     156
                     1574    1614*   1620    1660*   1666    1830*   1836*   1842    1902*   1908    1948*   1954     199
                     2000    2040*   2046    2132*   2138    2286*   2292*   2298    2430*   2436    2522*   2528     256
                     2574    2668*   2674*   2680    2740*   2746*   2752    2812*   2818*   2824    2978*   2984     302
                     3033*   3039    3100*   3106*   3112    3188*   3194*   3200    3296*   3302*   3308    3447*    346
                     3541*
$STUP = 177777        82#
$SVPC = 001000        87#     92
$SWR  = 160000        14      15#    3360    3367    3377    3386    3387    3664
$SWREG  000422       116#   3147    3255    3397    3399    3401    3403    3554    3558    3573    3576
$TESTN  000404       107#    225*    269     315     361     407     453     499     545     591     637     683      72
                      775     821     875     947    1019    1091    1163    1209    1255    1307    1353    1405     145
```

```
                                1497   1543   1589   1635   1681   1733   1805   1877   1923   1969   2015   2061   210
                                2153   2207   2261   2333   2405   2451   2497   2543   2589   2643   2715   2787   285
                                2937   3000   3073   3127   3235   3343   3399   3410*
$TN     = 000001                 14#
$TPB      000552                193#    221   3172*  3175*  3280*  3283*  3692*
$TPS      000550                192#  3159*  3165*  3173   3176*  3267*  3273*  3281   3284*  3690
$TSTM     000434                145#
$TTTLE    017240               3694#  3708
$TYPE     017170                210   3679#  3707
$UNIT     000412                110#
$UNITM    000440                147#
$USWR     000424                117#
.       = 017374                 81#    82     87     88#    90#    92#    96#    132    133#   135#   137#   149#   15
                                153#   155#   157#   159#   161#   199#   206#   250    255    260    265    270    29
                                301    306    311    316    342    347    352    357    362    388    393    398    40
                                408    439    444    449    454    485    490    495    500    526    531    536    54
                                546    572    577    582    587    592    618    623    628    633    638    664    66
                                674    679    684    715    720    725    730    761    766    771    776    802    80
                                812    817    822    851    856    861    866    871    876    908    913    918    92
                                928    933    938    943    948    985    990    995   1000   1005   1010   1015   102
                               1052   1057   1062   1067   1072   1077   1082   1087   1092   1129   1134   1139   114
                               1149   1154   1159   1164   1190   1195   1200   1205   1210   1236   1241   1246   125
                               1256   1293   1298   1303   1308   1334   1339   1344   1349   1354   1391   1396   140
                               1406   1432   1437   1442   1447   1452   1478   1483   1488   1493   1498   1524   152
                               1534   1539   1544   1570   1575   1580   1585   1590   1616   1621   1626   1631   163
                               1662   1667   1672   1677   1682   1719   1724   1729   1734   1771   1776   1781   178
                               1791   1796   1801   1806   1838   1843   1848   1853   1858   1863   1868   1873   187
                               1904   1909   1914   1919   1924   1950   1955   1960   1965   1970   1996   2001   200
                               2011   2016   2042   2047   2052   2057   2062   2093   2098   2103   2108   2134   213
                               2144   2149   2154   2183   2188   2193   2198   2203   2208   2237   2242   2247   225
                               2257   2262   2294   2299   2304   2309   2314   2319   2324   2329   2334   2371   237
                               2381   2386   2391   2396   2401   2406   2432   2437   2442   2447   2452   2483   248
                               2493   2498   2524   2529   2534   2539   2544   2570   2575   2580   2585   2590   261
                               2624   2629   2634   2639   2644   2676   2681   2686   2691   2696   2701   2706   271
                               2716   2748   2753   2758   2763   2768   2773   2778   2783   2788   2820   2825   283
                               2835   2840   2845   2850   2855   2860   2897   2902   2913   2918   2923   2928   293
                               2938   2965#  2966#  2974#  2975#  2980   2985   2991   2996   3001   3020#  3021#  303
                               3040   3045   3050   3055   3060   3065   3070   3074   3092#  3093#  3099#  3100#  310
                               3113   3118   3123   3128   3151   3196   3201   3206   3212   3217   3222   3227   323
                               3240#  3241#  3259   3304   3309   3314   3320   3325   3330   3335   3344   3348#  334
                               3387   3390#  3392#  3393#  3431#  3432#  3442#  3443#  3457#  3458#  3479#  3480#  349
                               3491#  3503#  3504#  3525#  3526#  3535#  3536#  3572   3574   3577   3600   3615   361
                               3623   3630#  3647   3666
.BIT    = 177777                  4#    35     36     37     38     39     40     41     42   3554   3558   3576
.PR       000544                190#  3599*  3601*  3602*  3616*  3617   3620*
.PRBUF    017020               3604   3622   3626   3630#
.PRF      016740               3606   3612#
.PRL      016726               3607#  3621
.$X     = 000430                132#   137
..A     = 016410               2965#  2974#  3020#  3092#  3099#  3240#  3348#  3431#  3442#  3457#  3479#  3490#  350
                               3525#  3535#
..B     = 016414               2965#  2966   2974#  2975   3020#  3021   3092#  3093   3099#  3100   3240#  3241   334
                               3349   3431#  3432   3442#  3443   3457#  3458   3479#  3480   3490#  3491   3503#  350
                               3525#  3526   3535#  3536
..C     = 000067               2965#  2974#  3020#  3092#  3099#  3240#  3348#  3431#  3442#  3457#  3479#  3490#  350
                               3525#  3535#
```

```
00100   CVKACC  MACY11 30A(1052)  21-AUG-78  15:28  PAGE 102
00200   CVKACC.P11    16-AUG-78 08:41          CROSS REFERENCE TABLE -- MACRO NAMES                                    SEQ 0108
00300
00400   COMMEN        1#
00500   DUMP         23#    3564    3591
00600   ENDCOM        1#
00700   ESCAPE        1#
00800   MFPS         19#    2973    3098    3441    3456    3489    3502    3534
00900   MTPS         21#    2964    3019    3091    3239    3347    3430    3478    3524
01000   MULT          1#
01100   NEWTST        1#
01200   POP           1#      4#    3654
01300   PRINT        25#
01400   PUSH          1#      4#    3638
01500   REPORT        1#
01600   SDUMP        24#
01700   SETUP         1#
01800   SKIP          1#
01900   SLASH         1#
02000   STARS         1#      4#      79      83      97     127     131     138     194    3354    3393    3552    3584    3631
02100   TYPBIN        1#
02200   TYPDEC        1#
02300   TYPNAM        1#
02400   TYPNUM        1#
02500   TYPOCS        1#
02600   TYPOCT        1#
02700   TYPTXT        1#
02800   $ADDER      194#     881    1025
02900   $ADDES      194#     953    1097
03000   $ADDR       194#     229     275     321     367     505     551     597     643     781
03100   $ADDS       194#     413     459     689     735
03200   $ADRER      194#    3005    3077
03300   $ALL        194#    2943
03400   $ALL2       194#    2969
03500   $DIVER      194#    2649    2721    2793
03600   $DIVES      194#    2865
03700   $DIVR       194#    2411    2503    2549
03800   $DIVS       194#    2457
03900   $FATL        17#     252     257     262     267     272     298     303     308     313     318     344     349     354     35
04000                364     390     395     400     405     410     435     441     446     451     456     481     487     492     49
04100                502     528     533     538     543     548     574     579     584     589     594     620     625     630     63
04200                640     666     671     676     681     686     711     717     722     727     732     757     763     768     77
04300                778     804     809     814     819     824     853     858     863     868     873     878     902     910     91
04400                920     925     930     935     940     945     950     972     981     987     992     997    1002    1007    101
04500               1017    1022    1046    1054    1059    1064    1069    1074    1079    1084    1089    1094    1116    1125    113
04600               1136    1141    1146    1151    1156    1161    1166    1192    1197    1202    1207    1212    1238    1243    124
04700               1253    1258    1289    1295    1300    1305    1310    1336    1341    1346    1351    1356    1387    1393    139
04800               1403    1408    1434    1439    1444    1449    1454    1480    1485    1490    1495    1500    1526    1531    153
04900               1541    1546    1572    1577    1582    1587    1592    1618    1623    1628    1633    1638    1664    1669    167
05000               1679    1684    1715    1721    1726    1731    1736    1758    1767    1773    1778    1783    1788    1793    179
05100               1803    1808    1832    1840    1845    1850    1855    1860    1865    1870    1875    1880    1906    1911    191
05200               1921    1926    1952    1957    1962    1967    1972    1998    2003    2008    2013    2018    2044    2049    205
05300               2059    2064    2089    2095    2100    2105    2110    2136    2141    2146    2151    2156    2185    2190    219
05400               2200    2205    2210    2239    2244    2249    2254    2259    2264    2288    2296    2301    2306    2311    231
05500               2321    2326    2331    2336    2358    2367    2373    2378    2383    2388    2393    2398    2403    2408    243
05600               2439    2444    2449    2454    2479    2485    2490    2495    2500    2526    2531    2536    2541    2546    257
05700               2577    2582    2587    2592    2621    2626    2631    2636    2641    2646    2670    2678    2683    2688    269
05800               2698    2703    2708    2713    2718    2742    2750    2755    2760    2765    2770    2775    2780    2785    279
05900               2814    2822    2827    2832    2837    2842    2847    2852    2857    2862    2884    2893    2899    2904    291
```

```
                2920    2925    2930    2935    2940    2982    2987    2993    2998    3003    3029    3037    3042    3047    305
                3057    3062    3067    3072    3076    3102    3110    3115    3120    3125    3130    3191    3198    3203    320
                3214    3219    3224    3229    3238    3299    3306    3311    3316    3322    3327    3332    3337    3346    354
$FINT            194#    3134    3242
$FIS7            194#     827    2159    2213    2595
$HLT              27#    3552
$MULER           194#    2267
$MULES           194#    2339
$MULR            194#    1883    1929    1975    2021    2113
$MULS            194#    2067
$OCTAL            28#    3599
$SCOPE            26#    3393
$SUBER           194#    1811
$SUBES           194#    1739
$SUBR            194#    1169    1215    1313    1411    1457    1503    1549    1595    1641
$SUBS            194#    1261    1359    1687
$SWDOC            22#      30
$$ESCA             1#
$$FATL            18#     252     257     262     267     272     298     303     308     313     318     344     349     354     35
                 364     390     395     400     405     410     435     441     446     451     456     481     487     492     49
                 502     528     533     538     543     548     574     579     584     589     594     620     625     630     63
                 640     666     671     676     681     686     711     717     722     727     732     757     763     768     77
                 778     804     809     814     819     824     853     858     863     868     873     878     902     910     91
                 920     925     930     935     940     945     950     972     981     987     992     997    1002    1007    101
                1017    1022    1046    1054    1059    1064    1069    1074    1079    1084    1089    1094    1116    1125    113
                1136    1141    1146    1151    1156    1161    1166    1192    1197    1202    1207    1212    1238    1243    124
                1253    1258    1289    1295    1300    1305    1310    1336    1341    1346    1351    1356    1387    1393    139
                1403    1408    1434    1439    1444    1449    1454    1480    1485    1490    1495    1500    1526    1531    153
                1541    1546    1572    1577    1582    1587    1592    1618    1623    1628    1633    1638    1664    1669    167
                1679    1684    1715    1721    1726    1731    1736    1758    1767    1773    1778    1783    1788    1793    179
                1803    1808    1832    1840    1845    1850    1855    1860    1865    1870    1875    1880    1906    1911    191
                1921    1926    1952    1957    1962    1967    1972    1998    2003    2008    2013    2018    2044    2049    205
                2059    2064    2089    2095    2100    2105    2110    2136    2141    2146    2151    2156    2185    2190    219
                2200    2205    2210    2239    2244    2249    2254    2259    2264    2288    2296    2301    2306    2311    231
                2321    2326    2331    2336    2358    2367    2373    2378    2383    2388    2393    2398    2403    2408    243
                2439    2444    2449    2454    2479    2485    2490    2495    2500    2526    2531    2536    2541    2546    257
                2577    2582    2587    2592    2621    2626    2631    2636    2641    2646    2670    2678    2683    2688    269
                2698    2703    2708    2713    2718    2742    2750    2755    2760    2765    2770    2775    2780    2785    279
                2814    2822    2827    2832    2837    2842    2847    2852    2857    2862    2884    2893    2899    2904    291
                2920    2925    2930    2935    2940    2982    2987    2993    2998    3003    3029    3037    3042    3047    305
                3057    3062    3067    3072    3076    3102    3110    3115    3120    3125    3130    3191    3198    3203    320
                3214    3219    3224    3229    3238    3299    3306    3311    3316    3322    3327    3332    3337    3346    354
$$NEWT             1#
$$SKIP             1#
.EQUAT             1#
.HEADE             1#      4#
.KT11              1#
.SETUP             1#      4#      82
.SWRHI             1#
.$ACT1             1#      4#      83
.$APTB             1#      4#      97
.$APTH             1#      4#     127
.$APTV             1#
.$ASTA             1#
.$CATC             1#
.$CMTA             1#
```

```
00100   CVKACC  MACY11 30A(1052)  21-AUG-78  15:28  PAGE 104
00200   CVKACC.P11    16-AUG-78 08:41          CROSS REFERENCE TABLE -- MACRO NAMES                    SEQ 0110
00300
00400   .$DB2D      1#
00500   .$DB2O      1#
00600   .$DIV       1#
00700   .$EOP       1#      4#    3354
00800   .$ERRO      1#
00900   .$ERRT      1#
01000   .$MULT      1#
01100   .$POWE      1#      4#    3631
01200   .$RAND      1#
01300   .$RDDE      1#
01400   .$RDOC      1#
01500   .$READ      1#
01600   .$R2AZ      1#
01700   .$SAVE      1#
01800   .$SB2D      1#
01900   .$SB2O      1#
02000   .$SCOP      1#
02100   .$SIZE      1#
02200   .$SUPR      1#
02300   .$TRAP      1#
02400   .$TYPB      1#
02500   .$TYPD      1#
02600   .$TYPE      1#
02700   .$TYPO      1#
02800   .$40CA      1#
02900
03000
03100   . ABS.  017374      000
03200
03300
03400     ERRORS DETECTED:  0
03500
03600     CVKACC.BIN,CVKACC.LST/CRF/SOL=CVKACC.SML,CVKACC.P11
03700     RUN-TIME: 17 23 1 SECONDS
03800     RUN-TIME RATIO: 575/42=13.4
03900     CORE USED:  41K  (81 PAGES)
04000
04100
```