FP11

BASIC INSTRUCTION EXERCISER

MD-11-DCFPO-D

EP-DCFPO-D-DL

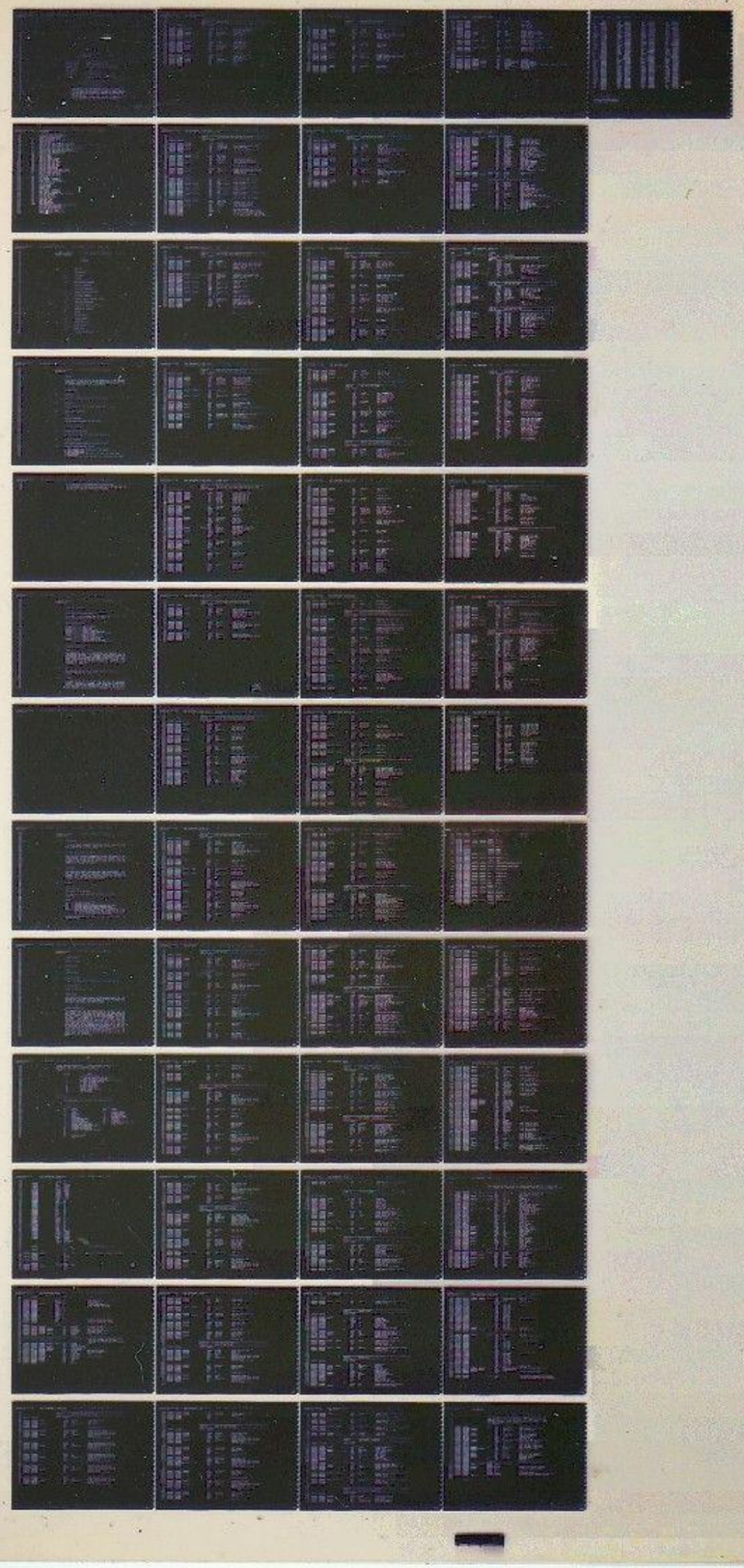
COPYRIGHT © 72-75

FICHE 1 OF 1

MAY 1978

digital

MADE IN USA



```
                        .REPT   0
```

[redacted]

IDENTIFICATION

PRODUCT CODE: MAINDEC-11-DCFPO-D-D

PRODUCT NAME: FP11 BASIC INSTRUCTION EXERCISER

DATE CREATED: JANUARY 1975

MAINTAINER: DIAGNOSTIC GROUP

AUTHORS: BOB BRAIN/STANLEY HARACKIEWICZ

COPYRIGHT: (C) 1972,1973,1975

DIGITAL EQUIPMENT CORPORATION

THIS MATERIAL IN THIS DOCUMENT IS FOR INFORMATION
PURPOSES ONLY AND IS SUBJECT TO CHANGE WITHOUT NOTICE.
DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY
FOR THE USE OF SOFTWARE ON EQUIPMENT WHICH IS NOT
SUPPLIED BY IT.
DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY
FOR ANY ERRORS WHICH MAY APPEAR IN THE DOCUMENT.

TABLE OF CONTENTS

CONTENTS

DESCRIPTION

1. ABSTRACT

THIS PROGRAM TESTS THE FP11 IN ALL MODES WITH FIXED NUMBER PATTERNS. IT RUNS WITH INTERUPTS BOTH ENABLED AND DISABLED AND CAUSES ERROR CONDITIONS. THE PROGRAM SHOULD BE RUN FOR AT LEAST 2 PASSES WITH ALL SWITCHES DOWN.

2. REQUIREMENTS

2.1 EQUIPMENT

PDP11/45 STANDARD COMPUTER WITH FP11 OPTION

2.2 STORAGE

PROGRAM STORAGE - THE ROUTINES USE MEMORY 0 - 17776

2.3 PRELIMINARY PROGRAMS

NONE

3. LOADING PROCEDURE

USE STANDARD PROCEDURE FOR ABS TAPES.

4. STARTING PROCEDURE

4.1 CONTROL SWITCH SETTINGS

SEE 5.1.1 (ALL DOWN FOR WORST CASE TESTING)

4.2 STARTING ADDRESS

THE PROGRAM SHOULD ALWAYS BE STARTED AT 200.

4.3 PROGRAM AND/OR OPERATOR ACTION

1) LOAD PROGRAM INTO MEMORY USING ABS LOADER.
2) LOAD ADDRESS 200.
3) SET SWITCHES (SEE SEC 5.1.1) ALL DOWN FOR WORST CASE
4) PRESS START.
5) THE PROGRAM WILL LOOP AND BELL WILL RING ONCE EVERY PASS

```
                                        6) A MINIMUM OF TWO PASSES SHOULD ALWAYS BE RUN,
                                        7) THE DISPLAY ON THE 11/45 WILL SHOW THE ITERATION COUNT IN
                                        THE LEFT BYTE AND TEST NUMBER IN THE RIGHT.  TO USE, SET THE
                                        DATA DISPLAY SWITCH TO THE DISPLAY POSITION.
```

5. OPERATING PROCEDURE

5.1 OPERATIONAL SWITCH SETTINGS

AT SA 200 .. ALL SWITCHES DOWN IS WORST CASE TESTING. THE BELL WILL RING UPON COMPLETION OF A PASS.

5.1.1 SWITCH SETTINGS ARE!

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

5.2 SUBROUTINE ABSTRACTS

5.2.1 SCOPE

THIS SUBROUTINE CALL IS PLACED BETWEEN EACH SUBTEST IN THE INSTRUCTION SECTION. IT RECORDS THE STARTING ADDRESS OF EACH SUBTEST AS IT IS BEING ENTERED IN LOCATION "LAD". IF A SCOPE LOOP IS REQUESTED, THE CURRENT SUBTEST WILL BE LOOPED UPON. SW<11> ON A 1 INHIBITS ITERATION OF SUBTESTS. THE CONTENTS OF LAD MAY BE USED TO DETERMINE THE LAST SUBTEST SUCCESSFULLY COMPLETED.

5.2.2 HLT

THIS ROUTINE PRINTS OUT AN ERROR MESSAGE (SEE 6.1). TO INHIBIT TYPEOUTS, PUT SW<13> ON A 1.

5.2.3 TRTRAP

IF SW<12> IS ON A 0, THE T BIT WILL BE SET ON ALTERNATE PASSES. WHEN SET, IT CAUSES A TRAP 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

216 REACHED.

5.2.4 TRAPCATCHER

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

5.2.5 FLOATING POINT TRAP (TO 244)

WHEN THE FP11 INTERRUPT DISABLE BIT IS SET, NO TRAPS TO 244 SHOULD OCCUR. IF AN INTERRUPT OCCURS, THE PROGRAM WILL HALT AT 766 IN THE ROUTINE CALLED FLTERR AND DISPLAY THE FPS REGISTER IN R0. WHEN THE INTERUPT DISABLE BIT IS CLEAR, THE PROGRAM WILL HANDLE THE TRAPS.

5.2.6 RELOCATOR

WITH SW<9> ON A ZERO, FLOATING POINT TEST (STARTING AT TEST 40) ARE RELOCATED AND EXECUTED THROUGHOUT MEMORY IN 1 K BLOCKS. WHEN AN ERROR IS ENCOUNTED, THE ACTUAL ADDRESS IS PRINTED OUT. TO FIND THE LISTING ADDRESS, MASK OUT BITS 12 THROUGH 15 AND ADD 10000 TO THE RESULT.

6. ERRORS

6.1 ERROR PRINTOUT

THE FORMAT IS AS FOLLOWS:

```
ADR  FPS  ANS1 ANS2 ANS3 ANS4 ANS5 ANS6 ANS7 ANS8
  FEC FEA
```

WHERE:

ADR = ADDRESS OF ERROR HLT
FPS = FLOATING POINT STATUS OR BAD DATA IN "FPS"
FEC = FLOATING EXCEPTION CODES (ERROR CODES)
FEA = FLOATING EXCEPTION ADDRESS (ERROR ADDRESS)
ANS1-8 = ERROR DATA READ FROM THE FP11. FROM 0-8 OF THESE MAY BE TYPED DEPENDING ON THE NUMBER FOLLOWING THE HLT; I.E., HLT+3 WOULD TYPE ANS1-ANS3.

TO FIND THE FAILING TEST, LOOK AT THE LISTING ABOVE THE ADDRESS TYPED.

6.2 ERROR RECOVERY

RESTART AT 200

7. RESTRICTIONS

NONE

8. MISCELLANEOUS

8.1 EXECUTION TIME

A BELL WILL RING WITHIN 1MINUTE WITH ALL SWITCHES DOWN.

8.2 STACK POINTER

STACK IS INITALLY SET TO 600

8.3 POWER FAIL

THIS TEST CAN BE POWER FAILED WITH NO ERRORS EXCEPT ON THE FEC AND FEA. TO USE, START THE TEST AS USUAL AND POWER DOWN THEN UP AT ANY TIME. THE PROGRAM SHOULD TYPE "POWER" AND CONTINUE TO RUN WITH NO OTHER TYPEOUTS.

9. PROGRAM DESCRIPTION

THIS PROGRAM TEST ALL THE INSTRUCTIONS ON THE FP11 AND ALL THE MODES. IT IS DIVIDED INTO SUBTESTS (THE CODE BETWEEN 2 SCOPE STATEMENTS) WHICH ARE RUN 256 TIMES BEFORE CONTINUING TO THE NEXT. SW<11> ON A 1 CAUSES EACH SUBTEST TO BE RUN ONLY ONCE. WITH SW<9> ON A 1, THE PROGRAM WILL NOT RELOCATE. THE ADDRESS ICNT (LOC 1000) AND DISPLAY REGISTER ON THE 11/45 EACH CONTAIN THE ITERATION COUNT IN THE LEFT BYTE AND THE TEST NUMBER IN THE RIGHT BYTE. ALL THE SUBTESTS SHOULD BE RUN SEQUENTIALLY BY STARTING AT 200 NOT BY STARTING AT THE BEGINNING OF THE SUBTEST. TO LOOP ON A PARTICULAR SUBTEST, PUT THE TEST NUMBER (SEE LISTING) IN THE RIGHT BYTE OF THE SWITCH REGISTER AND SW<8> ON A 1. THIS TEST WILL BE LOOPED UPON UNTIL SW<8> IS PUT ON A 0 OR THE RIGHT BYTE IS CHANGED. IF THE TEST IS NON-EXISTANT, THE PROGRAM WILL BE RUN AS USUAL.

.ENDR

```
                        .TITLE  MAINDEC-11-DCFPO-D      FP11 EXERCISER  STAND ALONE
                        ;COPYRIGHT 1972, DIGITAL EQUIPMENT CORP., MAYNARD, MASS
                        ;PROGRAM BY BOB BRAIN AND STAN HARACKIEWICZ
                        .REM*

                                SWITCH                  USE
                                ------          --------------------
                                  8             0 - LOAD UB REGISTER WITH SW<7:0>
                                                1 - LOOP ON TEST IN SW<7:0>
                                  9             RELOCATE * EXECUTE TESTS THROUGHOUT MEMORY
                                 10             0 - BELL ON PASS COMPLETE
                                                1 - BELL ON ERROR
                                 11             INHIBIT ITERATIONS
                                 12             INHIBIT TRACE TRAP
                                 13             INHIBIT ERROR TYPEOUTS
                                 14             LOOP ON TEST
                                 15             HALT ON ERROR


                        OUTPUT FORMI

                                ADR  FPS  ANS1 ANS2 ANS3 ANS4 ANS5 ANS6 ANS7 ANS8
                                  FEC  FEA


                                    FPS                                     FEC
                            BIT     REASON                          CODE    ERROR
                            ---     ------                          ----    -----
                             0      CARRY                            0      ADDRESS ERROR
                             1      OVERFLOW                         2      OPCODE ERROR
                             2      ZERO                             4      DIVIDE BY ZERO
                             3      NEGATIVE                         6      CONVERSION ERROR
                             4      MAINTAINANCE MODE               10      OVERFLOW
                             5      TRUNCATE MODE                   12      UNDERFLOW
                             6      LONG INTEGER MODE               14      UNDEFINED VARIABLE (=0)
                             7      DOUBLE PRECISION MODE           16      UBREAK TRAP
                             8      INTERUPT ON CONVERSION ERROR
                             9      INTERUPT ON OVERFLOW
                            10      INTERUPT ON UNDERFLOW
                            11      INTERUPT ON UNDEFINED VARIABLE
                            12
                            13
                            14      INTERUPT DISABLE
                            15      ERROR FLAG*
```

```
                                        .ENABL  ABS
                        000001          N=      1
                        177776          PS=     177776
                        177570          SWR=    177570
                        040000          SW14=   40000
                        020000          SW13=   20000
                        010000          SW12=   10000
                        004000          SW11=    4000
                        002000          SW10=    2000
                        001000          SW09=    1000
                        000400          SW08=     400
                        000000          FPS=    %0
                        000000          R0=     %0
                        000001          RETURN= %1
                        000001          R1=     %1
                        000002          R2=     %2
                        000003          R3=     %3
                        000004          R4=     %4
                        000005          R5=     %5
                        000005          TTY=    %5
                        000006          SP=     %6
                        000007          PC=     %7
                        000004          TYPE=   IOT
                        000007          BELL=   7
                        000000          AC0=    %0
                        000001          AC1=    %1
                        000002          AC2=    %2
                        000003          AC3=    %3
                        000004          AC4=    %4
                        000005          AC5=    %5
                        170003          LDUB=   170003
                        170005          STAB=   170005
                        170007          STOB=   170007
                        170006          MRS=    170006
                        170004          LDSC=   170004
                        177776          PS=     177776
                                        .EQUIV  TRAP,SCOPE
                                        .EQUIV  EMT,HLT

                        000000          .=      0                       ;TRAP CATCHER FROM 0 - 776
(1)
(1)             000200                  .=      200
(1)
(1)     000200  000167  000632                  JMP     START           ;JUMP TO STARTING ADDRESS OF PROGRAM
                000046                  .=      46
000046  011406                          LOGICAL
                000052                  .=      52
000052  040000                          40000

                000760                  .=      760
000760  170200                  FLTERR: STFPS   FPS
000762  170367  000044                  STST    FEC
000766  000000                          HALT
000770  000002                          RTI
```

```
                001000                  .=      1000
                
001000  000000                  ICNT:   0               ;ITERATION COUNT - LH   TEST NO. - RH
001002  000000                  ERRORS: 0               ;ERROR COUNT
001004  000000  000000          PCNT:   0,0             ;2 WORD PASS COUNT
001010  000000                  LAD:    0               ;LOOP ADDRESS FOR SCOPE
001012  000000                  ANS1:   0               ;FIRST ANSWER (SEE CODE)
001014  000000                  ANS2:   0
001016  000000                  ANS3:   0
001020  000000                  ANS4:   0
001022  000000                  ANS5:   0
001024  000000                  ANS6:   0
001026  000000                  ANS7:   0
001030  000000                  ANS8:   0
001032  000000                  FEC:    0               ;FLOATING EXCEPTION CODES
001034  000000                  FEA:    0               ;FLOATING EXECPTION ADDRESS

001036                          START:
001036  012706  000600                  MOV     #600,SP         ;** STACK AT 600 **
001042  012737  000002  000016          MOV     #2,@#16         ;SET RTI INTO 16
001050  012737  001072  000004          MOV     #M1120,@#4      ;FIND OUT WHICH MACHINE THIS IS
001056  005737  177772                  TST     @#177772        ;IS PIRQ THERE?
001062  012737  000006  000016          MOV     #6,@#16         ;FUDGE IN RTT IF 11/45
001070  000403                          BR      BEGIN

001072  016737  011542  000010  M1120: MOV     FPTADR,@#10     ;LOAD THE ILLEGAL INSTRUCTION VECTOR
                                                                ; WITH THE ADDRESS OF THE FPU,
                                                                ; THE FPU WILL HANDLE THE BAD OPCODES
001100  012737  000006  000004  BEGIN: MOV     #6,@#4          ;RESET 4
001106  012706  000600                  MOV     #600,SP
001112  012777  012306  011504          MOV     #POWDWN,@DWNVEC
001120  012777  000340  011500          MOV     #340,@DWNVEC+2
001126  012737  012506  000020          MOV     #IOTS,@#20      ;SET UP VECTOR 20
001134  012700  000030                  MOV     #30,R0          ;SET R0 TO VECTOR 30
001140  012720  011560                  MOV     #.,TRP:(0)+     ;SET EMT VECTOR
001144  012720  000340                  MOV     #340,(0)+
001150  012720  011420                  MOV     #.,EMT:(0)+     ;SET TRAP VECTOR
001154  012710  000340                  MOV     #340,(0)
001160  005067  177614                  CLR     ICNT
001164  005067  177620                  CLR     LAD
001170  005001                          CLR     RETURN
```

```
 566                                    ;*****************************************************************
 (2)                                    ;TEST 1          TEST OF WRITABILITY OF FPS
 (2)                                    ;*****************************************************************
 (2)  001172  104400                    TST1:   SCOPE
 567
 568  001174  170127  147757                   LDFPS   #147757         ;TEST FPS WITH ALL 1'S
 569  001200  170200                           STFPS   FPS             ;GET STATUS
 570  001202  022700  147757                   CMP     #147757,FPS     ;CHECK IT
 571  001206  001401                           BEQ     .+4
 572  001210  104000                           HLT                     ;FPS NOT 147777
 573
 574  001212  170127  040000                   LDFPS   #40000          ;TEST FPS WITH 40000
 575  001216  170200                           STFPS   FPS             ;STORE FLOATING POINT STATUS
 (1)  001220  022700  040000                   CMP     #40000,FPS      ;CHECK FLOATING POINT STATUS
 (1)  001224  001401                           BEQ     .+4             ;BRANCH IF OK
 (1)  001226  104000                           HLT                     ;FPS NOT EQUAL TO 40000
 (1)
 576
 577  001230  170011                           SETD                    ;TEST OF DOUBLE BIT ON A 1
 578  001232  170200                           STFPS   FPS             ;STORE FLOATING POINT STATUS
 (1)  001234  022700  040200                   CMP     #40200,FPS      ;CHECK FLOATING POINT STATUS
 (1)  001240  001401                           BEQ     .+4             ;BRANCH IF OK
 (1)  001242  104000                           HLT                     ;FPS NOT EQUAL TO 40200
 (1)
 579
 580  001244  170001                           SETF                    ;TEST OF DOUBLE BIT ON A 0
 581  001246  170200                           STFPS   FPS             ;STORE FLOATING POINT STATUS
 (1)  001250  022700  040000                   CMP     #40000,FPS      ;CHECK FLOATING POINT STATUS
 (1)  001254  001401                           BEQ     .+4             ;BRANCH IF OK
 (1)  001256  104000                           HLT                     ;FPS NOT EQUAL TO 40000
 (1)
 582
 583  001260  170012                           SETL                    ;TEST OF LONG BIT ON A 1
 584  001262  170200                           STFPS   FPS             ;STORE FLOATING POINT STATUS
 (1)  001264  022700  040100                   CMP     #40100,FPS      ;CHECK FLOATING POINT STATUS
 (1)  001270  001401                           BEQ     .+4             ;BRANCH IF OK
 (1)  001272  104000                           HLT                     ;FPS NOT EQUAL TO 40100
 (1)
 585
 586  001274  170002                           SETI                    ;TEST OF LONG BIT ON A 0
 587  001276  170200                           STFPS   FPS             ;STORE FLOATING POINT STATUS
 (1)  001300  022700  040000                   CMP     #40000,FPS      ;CHECK FLOATING POINT STATUS
 (1)  001304  001401                           BEQ     .+4             ;BRANCH IF OK
 (1)  001306  104000                           HLT                     ;FPS NOT EQUAL TO 40000
 (1)
```

```
589                                     ;*****************************************************************
(2)                                     ;TEST 2         TEST OF CFCC
(2)                                     ;*****************************************************************
(2)  001310  104400                     TST2:   SCOPE
592
591  001312  170127  040017                       LDFPS   #40017          ;LOAD ALL STATUS BITS TO 1'S
592  001316  170000                       CFCC                            ;GET THEM INTO PS
593  001320  013700  177776               MOV     @#PS,FPS                ;GET THEM FOR TYPING
594  001324  042700  177760               BIC     #177760,FPS             ;CLEAR JUNK
595  001330  022700  000017               CMP     #17,FPS                 ;ALL SET?
596  001334  001401                       BEQ     .+4
597  001336  104000                       HLT                             ;PS NOT 17
598
599  001340  170127  040012               LDFPS   #40012          ;LOAD FPS WITH 12
600  001344  170000                       CFCC                            ;GET INTO PS
601  001346  013700  177776               MOV     @#PS,FPS                ;GET FOR TYPING
602  001352  042700  177760               BIC     #177760,FPS             ;CLEAR JUNK
603  001356  022700  000012               CMP     #12,FPS                 ;SAME AS LD?
604  001362  001401                       BEQ     .+4
605  001364  104000                       HLT                             ;PS NOT 12
606
607  001366  170127  040005               LDFPS   #40005          ;LOAD FPS WITH 5
608  001372  170000                       CFCC                            ;GET BITS
609  001374  013700  177776               MOV     @#PS,FPS                ;SET FOR TYPING
610  001400  042700  177760               BIC     #177760,FPS             ;CLEAR JUNK
611  001404  022700  000005               CMP     #5,FPS                  ;SAME?
612  001410  001401                       BEQ     .+4
613  001412  104000                       HLT                             ;PS NOT 5
```

```
615                                     ;*****************************************************************
(2)                                     ;TEST 3         TEST OF LDD, STD, CMPD OF -1,0,-1,0
(2)                                     ;*****************************************************************
(2)  001414  104400                     TST3:   SCOPE
616
617  001416  170127  047600                    LDFPS   #47600
618  001422  012700  011102                    MOV     #D1010,FPS      ;GET ADDRESS OF DATA WORD
619  001426  172420                            LDD     (FPS)+,AC0      ;LOAD INTO AC0
620  001430  022700  011112                    CMP     #D1010+10,FPS   ;IS THE NEW ADDRESS RIGHT?
621  001434  001401                            BEQ     .+4
622  001436  104000                            HLT                     ;FPS NOT D1010+10
623
624  001440  010667  011146                    MOV     SP,.TYPE
625  001444  162767  000010  011140           SUB     #10.,.TYPE
626  001452  174046                            STD     AC0,-(SP)       ;GET THE DATA BACK
627  001454  010600                            MOV     SP,FPS          ;SAVE THE SP FOR TYPING
628  001456  026706  011130                    CMP     .TYPE,SP        ;SP DECREMENTED PROPERLY?
629  001462  001401                            BEQ     .+4
630  001464  104000                            HLT                     ;SP NOT SP-10
631
632  001466  170200                            STFPS   FPS             ;STORE FLOATING POINT STATUS
(1)  001470  022700  047610                    CMP     #47610,FPS      ;CHECK FLOATING POINT STATUS
(1)  001474  001401                            BEQ     .+4             ;BRANCH IF OK
(1)  001476  104000                            HLT                     ;FPS NOT EQUAL TO 47610
(1)
633
634  001500  021667  007376                    CMP     (SP),D1010      ;CHECK FIRST POECE OF DATA
635  001504  001401                            BEQ     .+4
636  001506  104204                            HLT     204             ;DATA IN (SP) NOT D1010
637  001510  026667  000002  007366            CMP     2(SP),D1010+2   ;CHECK SECOND
638  001516  001401                            BEQ     .+4
639  001520  104204                            HLT     204             ;DATA IN 2(SP) NOT D1010+2
640  001522  026667  000004  007356            CMP     4(SP),D1010+4   ;CHECK THIRD
641  001530  001401                            BEQ     .+4
642  001532  104204                            HLT     204             ;DATA IN 4(SP) NOT D1010+4
643  001534  026667  000006  007346            CMP     6(SP),D1010+6   ;CHECK FOURTH
644  001542  001401                            BEQ     .+4
645  001544  104204                            HLT     204             ;DATA IN 6(SP) NOT D1010+6
646
647  001546  062767  000010  011036            ADD     #10.,.TYPE
648  001554  173426                            CMPD    (SP)+,AC0       ;RECHECK DATA AND SP
649  001556  010600                            MOV     SP,FPS          ;SAVE SP FOR TYPING
650  001560  026706  011026                    CMP     .TYPE,SP        ;CHECK ADDRESS IN SP
651  001564  001401                            BEQ     .+4
652  001566  104000                            HLT                     ;SP NOT RESTORED
653  001570  170200                            STFPS   FPS             ;GET FPS IN CASE CMPD FAILED
654  001572  170000                            CFCC                    ;NOW GET THE FP CONDITION CODES
655  001574  001401                            BEQ     .+4             ;IF IT HALTS HERE IT MUST BE THE
656  001576  104000                            HLT                     ; CMPD BECAUSE THE IS ALREADY CONFIRMED
```

```
                                ;*****************************************************************
                                ;TEST 4          TEST OF LDD, STD, CMPD OF 0,-1,0,-1
                                ;*****************************************************************
001600  104400          TST4:   SCOPE
001602  170127  047600          LDFPS   #47600
001606  172437  011104          LDD     @#D0101,AC0     ;LOAD 0,-1,0,-1 INTO AC0 *PIC*
001612  170200                  STFPS   FPS             ;STORE FLOATING POINT STATUS
001614  022700  047604          CMP     #47604,FPS      ;CHECK FLOATING POINT STATUS
001620  001401                  BEQ     .+4             ;BRANCH IF OK
001622  104000                  HLT                     ;FPS NOT EQUAL TO 47604
001624  012700  001012          MOV     #ANS1,FPS       ;ADDRESS TO BE STORED INTO
001630  174010                  STD     AC0,(FPS)       ;STORE IT INTO ANS1-4 *PIC*
001632  026767  007246  177152  CMP     D0101,ANS1      ;FIRST WORD OK?
001640  001401                  BEQ     .+4
001642  104004                  HLT     4               ;ANS1 NOT D0101
001644  026767  007236  177142  CMP     D0101+2,ANS2    ;SECOND
001652  001401                  BEQ     .+4
001654  104004                  HLT     4               ;ANS2 NOT D0101+2
001656  026767  007226  177132  CMP     D0101+4,ANS3    ;THIRD
001664  001401                  BEQ     .+4
001666  104004                  HLT     4               ;ANS3 NOT D0101+4
001670  026767  007216  177122  CMP     D0101+6,ANS4    ;FOURTH
001676  001401                  BEQ     .+4
001700  104004                  HLT     4               ;ANS4 NOT D0101+6

001702  012704  011126          MOV     #AD0101-2,R4    ;ADDRESS-2 OF DATA
001706  173474  000002          CMPD    @2(4),AC0       ;CHECK DATA IN AC0 *PIC*
001712  170200                  STFPS   FPS             ;GET STAUS FOR TYPING
001714  170000                  CFCC                    ;GET CONDITION CODES
001716  001401                  BEQ     .+4
001720  104000                  HLT                     ;CMPD FAILED
```

```
                                ;*****************************************************************
                                ;TEST 5         TEST OF LDF, STF, CMPF OF -1.0
                                ;*****************************************************************
001722  104400          TST5:   SCOPE

                        LDFPS   #47400          ;SET FLOATING MODE
001724  170127  047400  LDF     D1010,AC0       ;LOAD -1.0 INTO AC0
001730  172467  007146  MOV     #ANS1,FPS       ;POINTED TO ANSWER AREA *PIC*
001734  012700  001012  STF     AC0,(FPS)+      ;STORE RESULT
001740  174020          CMP     #ANS3,FPS       ;INCREMENTED PROPERLY
001742  022700  001016  BEQ     .+4
001746  001401          HLT                     ;FPS NOT ANS1+4
001750  104000

001752  026767  007124  177032  CMP     D1010,ANS1      ;CHECK FIRST WORD
001760  001401          BEQ     .+4
001762  104002          HLT     2               ;ANS1 NOT D1010
001764  026767  007114  177022  CMP     D1010+2,ANS2    ;SECOND
001772  001401          BEQ     .+4
001774  104002          HLT     2               ;ANS2 NOT D1010+2

001776  170011          SETD                    ;GO TO DOUBLE MODE
002000  012700  001022  MOV     #ANS5,FPS       ;ADDRESS OF DATA+10
002004  174040          STD     AC0,-(FPS)      ;GET DATA
002006  022700  001012  CMP     #ANS1,FPS       ;CHECK FOR PROPER DECREMENTATION
002012  001401          BEQ     .+4
002014  104000          HLT                     ;FPS NOT ANS1

002016  012700  011134  MOV     #AD1000,FPS     ;LOAD ADDRESS OF ADDRESS OF DATA
002022  173430          CMPD    @(FPS)+,AC0     ;CHECK THE DATA
002024  022700  011136  CMP     #AD1000+2,FPS   ;FPS GETS INCREMENTED BY 2
002030  001401          BEQ     .+4
002032  104000          HLT                     ;FPS NOT AD1001+2

002034  170200          STFPS   FPS             ;GET FPS
002036  170000          CFCC                    ;COPY CONDITION CODES
002040  001401          BEQ     .+4             ;EITHER CMPD FAILED OR THE
002042  104004          HLT     4               ;LDF MODIFIED RIGHT HALF
```

```
                                ;*****************************************************************
                                ;TEST 6          TEST OF LDF, STF, CMPF WITH <=> IN ALL AC'S
                                ;*****************************************************************
002044  104400          TST6:   SCOPE

002046  170127  047400          LDFPS   #47400          ;LOAD FLOATING MODE
002052  172427  140640          LDF     #-5,AC0         ;LOAD AC0 WITH -5
002056  172527  140200          LDF     #-1,AC1         ;LOAD AC1 WITH -1
002062  172627  000000          LDF     #0,AC2          ;LOAD AC2 WITH 0
002066  172727  040640          LDF     #5,AC3          ;LOAD AC3 WITH 5
002072  174004                  STF     AC0,AC4         ;LOAD AC4 WITH -5
002074  174305                  STF     AC3,AC5         ;LOAD AC5 WITH 5

002076  174067  176710          STF     AC0,ANS1        ;GET AC0
002102  173427  140640          CMPF    #-5,AC0         ;CHECK IT
002106  170200                  STFPS   FPS             ;GET STATUS
002110  170000                  CFCC                    ;GET CC
002112  001401                  BEQ     .+4
002114  104002                  HLT     2               ;AC0 NOT -5

002116  174137  001012          STF     AC1,@#ANS1      ;GET AC1
002122  173567  176664          CMPF    ANS1,AC1        ;CHECK IT
002126  170200                  STFPS   FPS             ;GET STATUS
002130  170000                  CFCC                    ;GET CC
002132  001401                  BEQ     .+4
002134  104002                  HLT     2               ;AC1 NOT -1
002136  012704  001012  ADANS1: MOV     #ANS1,R4        ;POINTER TO ANSWER AREA
002142  174214                  STF     AC2,(4)         ;PUT DATA INTO ANS1
002144  173667  176642          CMPF    ANS1,AC2        ;CHECK DATA
002150  170200                  STFPS   FPS             ;SET STATUS
002152  170000                  CFCC                    ;GET CC
002154  001401                  BEQ     .+4
002156  104002                  HLT     2               ;AC2 NOT 0

002160  174377  177754          STF     AC3,@ADANS1+2   ;PUT DATA INTO ANS1
002164  173767  176622          CMPF    ANS1,AC3        ;CHECK DATA
002170  170200                  STFPS   FPS             ;GET STATUS
002172  170000                  CFCC                    ;GET CC
002174  001401                  BEQ     .+4
002176  104002                  HLT     2               ;AC3 NOT 5

002200  173404                  CMPF    AC4,AC0         ;CHECK AC4 FOR -5
002202  170200                  STFPS   FPS             ;GET STATUS
002204  170000                  CFCC                    ;GET CC
002206  001401                  BEQ     .+4
002210  104000                  HLT
                                                        ;AC0 NOT AC4

002212  173705                  CMPF    AC5,AC3         ;CHECK AC5 FOR 5
002214  170200                  STFPS   FPS             ;GET STATUS
002216  170000                  CFCC                    ;GET CC
002220  001401                  BEQ     .+4
002222  104000                  HLT                     ;AC3 NOT AC5
```

```
                                ;*****************************************************************
                                ;TEST 7          TEST OF CMPF WITH DATA IN AC0-AC5
                                ;*****************************************************************
002224  104400                  TST7:   SCOPE

002226  170127  041000                  LDFPS   #41000          ;LOAD STATUS WITH 0
002232  173405                          CMPF    AC5,AC0         ;CMP 5 TO -5
002234  170200                          STFPS   FPS             ;STORE FLOATING POINT STATUS
002236  022700  041000                  CMP     #41000,FPS      ;CHECK FLOATING POINT STATUS
002242  001401                          BEQ     .+4             ;BRANCH IF OK
002244  104000                          HLT                     ;FPS NOT EQUAL TO 41000

002246  173704                          CMPF    AC4,AC3         ;CMP -5 TO 5
002250  170200                          STFPS   FPS             ;STORE FLOATING POINT STATUS
002252  022700  041010                  CMP     #41010,FPS      ;CHECK FLOATING POINT STATUS
002256  001401                          BEQ     .+4             ;BRANCH IF OK
002260  104000                          HLT                     ;FPS NOT EQUAL TO 41010

002262  173767  000650                  CMPF    DBIG,AC3        ;MAKE IT OVERFLOW
002266  170200                          STFPS   FPS             ;STORE FLOATING POINT STATUS
002270  022700  041000                  CMP     #41000,FPS      ;CHECK FLOATING POINT STATUS
002274  001401                          BEQ     .+4
002276  104000                          HLT                     ;FPS NOT 141002
```

```
                                ;*****************************************************************
                                ;TEST 10        TEST OF TSTF AND TSTD USING OLD AC0-AC5
                                ;*****************************************************************
002300  104400                  TST10:  SCOPE

002302  170127  040000                  LDFPS   #40000
002306  170501                          TSTF    AC1             ;TEST AC1 = -1
002310  170200                          STFPS   FPS             ;GET THE STATUS
002312  022700  040010                  CMP     #40010,FPS      ;CHECK STATUS
002316  001401                          BEQ     .+4
002320  104000                          HLT                     ;N BIT NOT SET

002322  170505                          TSTF    AC5             ;TEST AC5 = 5
002324  170200                          STFPS   FPS             ;GET STATUS
002326  022700  040000                  CMP     #40000,FPS      ;CHECK STATUS
002332  001401                          BEQ     .+4
002334  104000                          HLT                     ;NOT 0

002336  170502                          TSTF    AC2             ;TEST AC2 = 0
002340  170200                          STFPS   FPS             ;GET STATUS
002342  022700  040004                  CMP     #40004,FPS      ;CHECK STATUS
002346  001401                          BEQ     .+4
002350  104000                          HLT                     ;Z BIT NOT SET

002352  170527  177777                  TSTF    #-1             ;TEST FOR THE N BIT IN LINE
002356  000401                          BR      .+4             ;SHOULD GO HERE
002360  104000                          HLT                     ;INCREMENTED BY 4 NOT 2
002362  170200                          STFPS   FPS             ;GET STATUS
002364  022700  040010                  CMP     #40010,FPS      ;CHECK THE N BIT
002370  001401                          BEQ     .+4
002372  104000                          HLT                     ;N BIT NOT SET

002374  170011                          SETD                    ;SET DOUBLE MODE
002376  170527  000000                  TSTD    #0              ;TEST FOR Z BIT IN LINE
002402  000403                          BR      .+10            ;SHOULD GO HERE
002404  000000                          HALT                    ;NOT HERE
002406  000000                          HALT                    ;OR HERE
002410  000000                          HALT                    ;OR HERE
002412  170200                          STFPS   FPS             ;GET STATUS
002414  022700  040204                  CMP     #40204,FPS      ;CHECK STATUS
002420  001401                          BEQ     .+4
002422  104000                          HLT                     ;Z BIT NOT SET
```

```
828                                     ;*****************************************************************
(2)                                     ;TEST 11        TEST OF CLRX INSTRUCTIONS
(2)                                     ;*****************************************************************
(2)  002424  104400                     TST11:  SCOPE
829
830  002426  170127  040200                    LDFPS   #40200          ;DOUBLE MODE
831  002432  172467  006444                    LDD     D1010,AC0       ;LOAD A -1
832  002436  174867  176358                    STD     AC0,ANS1        ;PUT INTO ANS1
833  002442  170001                             SETF                    ;SET FLOATING
834  002444  170467  176342                    CLRF    ANS1            ;CLEAR IT OUT
835  002450  170200                             STFPS   FPS             ;STORE FLOATING POINT STATUS
(1)  002452  022700  040004                    CMP     #40004,FPS      ;CHECK FLOATING POINT STATUS
(1)  002456  001401                            BEQ     .+4             ;BRANCH IF OK
(1)  002460  104000                            HLT                     ;FPS NOT EQUAL TO 40004
(1)
836
837  002462  170567  176324                    TSTF    ANS1            ;TEST FOR ZERO
838  002466  170000                             CFCC                    ;GET CC
839  002470  001401                            BEQ     .+4
840  002472  104002                            HLT     2               ;AC0 NOT ZERO
841
842  002474  026767  006406  176314             CMP     D1010+4,ANS3    ;CHECK THIRD WORD
843  002502  001401                            BEQ     .+4
844  002504  104004                            HLT     4               ;ANS3 NOT -1
845  002506  026767  006376  176304             CMP     D1010+6,ANS4    ;CHECK FOURTH
846  002514  001401                            BEQ     .+4
847  002516  104004                            HLT     4               ;ANS4 NOT 0
848
849  002520  170011                             SETD
850  002522  170400                             CLRD    AC0             ;CLEAR THE REST OF AC0
851  002524  170200                             STFPS   FPS             ;STORE FLOATING POINT STATUS
(1)  002526  022700  040204                    CMP     #40204,FPS      ;CHECK FLOATING POINT STATUS
(1)  002532  001401                            BEQ     .+4             ;BRANCH IF OK
(1)  002534  104000                            HLT                     ;FPS NOT EQUAL TO 40204
(1)
852  002536  174867  176250                    STD     AC0,ANS1        ;STORE RESULT
853  002542  170500                             TSTD    AC0             ;DID IT CLEAR
854  002544  170000                             CFCC                    ;GET STATUS
855  002546  001401                            BEQ     .+4
856  002550  104004                            HLT     4               ;DID NOT CLEAR
857
858  002552  172467  006370                    LDD     DMZERO,AC0      ;LOAD A MINUS ZERO
859  002556  170400                             CLRD    AC0             ;CLEAR IT OUT
860  002560  170200                             STFPS   FPS             ;STORE FLOATING POINT STATUS
(1)  002562  022700  040204                    CMP     #40204,FPS      ;CHECK FLOATING POINT STATUS
(1)  002566  001401                            BEQ     .+4             ;BRANCH IF OK
(1)  002570  104000                            HLT                     ;FPS NOT EQUAL TO 40204
(1)
861  002572  174867  176214                    STD     AC0,ANS1        ;STORE RESULT
862  002576  170500                             TSTD    AC0             ;CHECK IT
863  002600  170000                             CFCC                    ;GET CC
864  002602  001401                            BEQ     .+4
865  002604  104004                            HLT     4               ;DID NOT CLEAR
```

```
 867                                    ;*****************************************************************
 (2)                                    ;TEST 12        TEST OF NEGX
 (2)                                    ;*****************************************************************
 (2)  002606  104400            TST12:  SCOPE
 868
 869  002610  170127  040000            LDFPS   #40000
 870  002614  172427  140640            LDF     #-5,AC0         ;LOAD AC0 WITH -5
 871  002620  170700                     NEGF    AC0             ;MAKE IT 5
 872  002622  170200                     STFPS   FPS             ;STORE FLOATING POINT STATUS
 (1)  002624  022700  040000            CMP     #40000,FPS      ;CHECK FLOATING POINT STATUS
 (1)  002630  001401                    BEQ     .+4             ;BRANCH IF OK
 (1)  002632  104000                    HLT                     ;FPS NOT EQUAL TO 40000
 (1)
 873
 874  002634  174067  176152            STF     AC0,ANS1        ;GET THE RESULT
 875  002640  173427  040640            CMPF    #5,AC0          ;CHECK THE RESULT
 876  002644  170000                     CFCC                    ;GET CC
 877  002646  001401                    BEQ     .+4
 878  002650  104002                    HLT     2               ;RESULT NOT 5
 879
 880  002652  170767  176134            NEGF    ANS1            ;MAKE IT -5
 881  002656  170200                     STFPS   FPS             ;STORE FLOATING POINT STATUS
 (1)  002660  022700  040010            CMP     #40010,FPS      ;CHECK FLOATING POINT STATUS
 (1)  002664  001401                    BEQ     .+4             ;BRANCH IF OK
 (1)  002666  104000                    HLT                     ;FPS NOT EQUAL TO 40010
 (1)
 882
 883  002670  022767  140640  176114      CMP     #140640,ANS1    ;CHECK THE RESULT
 884  002676  001401                    BEQ     .+4
 885  002700  104002                    HLT     2               ;RESULT NOT -5
 886
 887  002702  005767  176106            TST     ANS2            ;TEST IT
 888  002706  001401                    BEQ     .+4             ;SKIP IF OK
 889  002710  104002                    HLT     2
 890
 891  002712  170127  047400            LDFPS   #47400          ;TURN ON INTERUPTS
 892  002716  170400                     CLRF    AC0             ;CLEAR AC0
 893  002720  170700                     NEGF    AC0             ;NEGATE IT
 894  002722  170200                     STFPS   FPS             ;STORE FLOATING POINT STATUS
 (1)  002724  022700  047404            CMP     #47404,FPS      ;CHECK FLOATING POINT STATUS
 (1)  002730  001401                    BEQ     .+4             ;BRANCH IF OK
 (1)  002732  104000                    HLT                     ;FPS NOT EQUAL TO 47404
 (1)
 895
 896  002734  174067  176052            STF     AC0,ANS1        ;GET RESULT
 897  002740  170500                     TSTF    AC0             ;CHECK IT
 898  002742  170000                     CFCC                    ;GET CC
 899  002744  001401                    BEQ     .+4
 900  002746  104002                    HLT     2               ;RESULT NOT 0
 901
 902  002750  170011                     SETD                    ;SET DOUBLE MODE
 903  002752  170400                     CLRD    AC0             ;CLEAR AC0
 904  002754  170700                     NEGD    AC0             ;NEGATE AC0
 905  002756  170200                     STFPS   FPS             ;STORE FLOATING POINT STATUS
```

```
    (1)  002760  022700  047604          CMP     #47604,FPS      ;CHECK FLOATING POINT STATUS
    (1)  002764  001401                  BEQ     .+4             ;BRANCH IF OK
    (1)  002766  104000                  HLT                     ;FPS NOT EQUAL TO 47604
    (1)
    926
    927  002770  174067  176016          STF     AC0,ANS1        ;GET RESULT
    928  002774  170500                  TSTF    AC0             ;TEST RESULT
    929  002776  170000                  CFCC                    ;GET CC
    930  003000  001401                  BEQ     .+4
    931  003002  104004                  HLT     4               ;RESULT NOT 0
    932
    933                          ;*****************************************************************
    (2)                          ;TEST 13        TEST OF ABSX
    (2)                          ;*****************************************************************
    (2)  003004  104400          TST13:  SCOPE
    934
    935  003006  170127  040000          LDFPS   #40000
    936  003012  172427  140200          LDF     #-1,AC0         ;LOAD A -1
    937  003016  174067  175770          STF     AC0,ANS1        ;GET IT
    938  003022  170667  175764          ABSF    ANS1            ;MAKE IT A 1
    939  003026  170200                  STFPS   FPS             ;STORE FLOATING POINT STATUS
    (1)  003030  022700  040000          CMP     #40000,FPS      ;CHECK FLOATING POINT STATUS
    (1)  003034  001401                  BEQ     .+4             ;BRANCH IF OK
    (1)  003036  104000                  HLT                     ;FPS NOT EQUAL TO 40000
    (1)
    940
    941  003040  022767  040200  175744  CMP     #40200,ANS1     ;CHECK FOR A 1
    942  003046  001401                  BEQ     .+4
    943  003050  104002                  HLT     2               ;RESULT NOT 1
    944
    945  003052  005767  175736          TST     ANS2            ;TEST IT
    946  003056  001401                  BEQ     .+4             ;SKIP IF OK
    947  003060  104002                  HLT     2
    948
    949  003062  170600                  ABSF    AC0             ;ABS IT AGAIN
    950  003064  170200                  STFPS   FPS             ;GET STATUS
    951  003066  173427  040200          CMPF    #1,AC0          ;CHECK FOR A 1
    952  003072  170000                  CFCC                    ;GET CC
    953  003074  001401                  BEQ     .+4
    954  003076  104002                  HLT     2               ;RESULT NOT 1
    955
    956  003100  172407  000042          LDF     DMZERO,AC0      ;LOAD A MINUS ZERO
    957  003104  170600                  ABSF    AC0             ;ABS IT
    958  003106  170200                  STFPS   FPS             ;STORE FLOATING POINT STATUS
    (1)  003110  022700  040004          CMP     #40004,FPS      ;CHECK FLOATING POINT STATUS
    (1)  003114  001401                  BEQ     .+4             ;BRANCH IF OK
    (1)  003116  104000                  HLT                     ;FPS NOT EQUAL TO 40004
    (1)
    959
    960  003120  174067  175666          STF     AC0,ANS1        ;GET RESULT
    961  003124  170500                  TSTF    AC0             ;TEST FOR 0
    962  003126  170000                  CFCC                    ;GET CC
    963  003130  001401                  BEQ     .+4
    964  003132  104002                  HLT     2               ;RESULT NOT 0
```

```
                                                ;SET DOUBLE MODE
                        LDD     D1010,AC0       ;LOAD -1.0,-1.0
                        ABSD    AC0             ;ABS IT
                        STFPS   FPS             ;STORE FLOATING POINT STATUS
                        CMP     #40200,FPS      ;CHECK FLOATING POINT STATUS
                        BEQ     .+4             ;BRANCH IF OK
                        HLT                     ;FPS NOT EQUAL TO 40200

                        STD     AC0,ANS1        ;GET RESULT
                        CMPD    DBIG,AC0        ;CHECK THE RESULT
                        CFCC                    ;GET CC
                        BEQ     .+4
                        HLT     4               ;RESULT NOT 77777,0.-1.0

;*****************************************************************
;TEST 14         TEST OF LDEXP & STEXP
;*****************************************************************
TST14:  SCOPE

                        LDFPS   #40200          ;SET DOUBLE MODE
                        LDD     D1010,AC0       ;LOAD A -1.0,-1.0
                        LDEXP   #-200,AC0       ;CLEAR THE EXPONENT
                        STFPS   FPS             ;STORE FLOATING POINT STATUS
                        CMP     #40204,FPS      ;CHECK FLOATING POINT STATUS
                        BEQ     .+4             ;BRANCH IF OK
                        HLT                     ;FPS NOT EQUAL TO 40204

                        STD     AC0,ANS1        ;GET THE RESULT
                        TSTD    AC0             ;IS IT 0
                        CFCC
                        BEQ     .+4
                        HLT     4               ;AC0 NOT 0

                        STEXP   AC0,ANS1        ;GET THE RESULT
                        MOV     @PS,FPS         ;GET PS BITS
                        BIC     #177700,FPS     ;CLEAR JUNK
                        CMP     #10,FPS         ;IS IT OK?
                        BEQ     .+4             ;SKIP IF OK
                        HLT                     ;PS IS WRONG

                        CMP     #-200,ANS1      ;CHECK IT
                        BEQ     .+4
                        HLT     1               ;EXPONENT NOT 0

                        SETF                    ;SET FLOATING MODE
                        LDF     D0101,AC0       ;LOAD A 0,-1
                        LDEXP   #200,AC0        ;SET EXPONENT TO -1
                        STFPS   FPS             ;STORE FLOATING POINT STATUS
                        CMP     #40000,FPS      ;CHECK FLOATING POINT STATUS
                        BEQ     .+4             ;BRANCH IF OK
                        HLT                     ;FPS NOT EQUAL TO 40000
```

```
  984
  985  003322  174067  175464          STF     AC0,ANS1        ;SAVE RESULT
  986  003326  005767  175460          TST     ANS1            ;CHECK FIRST WORD
  987  003332  001401                  BEQ     .+4
  988  003334  104002                  HLT     2               ;ANS1 NOT 0
  989
  990  003336  022767  177777  175458  CMP     #-1,ANS2        ;CHECK SECOND WORD
  991  003344  001401                  BEQ     .+4
  992  003346  104002                  HLT     2               ;ANS2 NOT -1
  993
  994  003350  175067  175436          STEXP   AC0,ANS1        ;GET THE EXPONENT BACK
  995  003354  022767  177600  175430  CMP     #-200,ANS1      ;CHECK IT
  996  003362  001401                  BEQ     .+4
  997  003364  104000                  HLT                     ;EXPONENT NOT -200
  998
  999  003366  176527  000052          LDEXP   #52,AC1         ;LOAD ALT 1'S
 1000  003372  175100                  STEXP   AC1,FPS         ;GET THEM BACK
 1001  003374  022700  000052          CMP     #52,FPS         ;OK?
 1002  003400  001401                  BEQ     .+4
 1003  003402  104000                  HLT                     ;EXP NOT 252
 1004
 1005  003404  176627  000025          LDEXP   #25,AC2         ;LOAD OTHER ALT 1'S
 1006  003410  175200                  STEXP   AC2,FPS         ;GET IT BACK
 1007  003412  022700  000025          CMP     #25,FPS         ;CHECK IT
 1008  003416  001401                  BEQ     .+4
 1009  003420  104000                  HLT                     ;EXP NOT 125
 1010
 1011                          ;*****************************************************************
  (2)                          ;TEST 15        TEST OF ADDF & SUBF
  (2)                          ;*****************************************************************
  (2)  003422  104400          TST15:  SCOPE
 1012
 1013  003424  170127  047400          LDFPS   #47400          ;LOAD FLOATING MODE
 1014  003430  172427  040600          LDF     #4,AC0          ;LOAD A 4
 1015  003434  172027  040400          ADDF    #2,AC0          ;ADD A 2
 1016  003440  170200                  STFPS   FPS             ;STORE FLOATING POINT STATUS
  (1)  003442  022700  047400          CMP     #47400,FPS      ;CHECK FLOATING POINT STATUS
  (1)  003446  001401                  BEQ     .+4             ;BRANCH IF OK
  (1)  003450  104000                  HLT                     ;FPS NOT EQUAL TO 47400
  (1)
 1017
 1018  003452  174067  175334          STF     AC0,ANS1        ;STORE RESULT
 1019  003456  173427  040700          CMPF    #6,AC0          ;CHECK RESULT
 1020  003462  170000                  CFCC                    ;GET CC
 1021  003464  001401                  BEQ     .+4
 1022  003466  104002                  HLT     2               ;RESULT NOT 6
 1023
 1024  003470  173027  041020          SUBF    #9,AC0          ;SUBTRACT 9 FROM 6
 1025  003474  170200                  STFPS   FPS             ;STORE FLOATING POINT STATUS
  (1)  003476  022700  047410          CMP     #47410,FPS      ;CHECK FLOATING POINT STATUS
  (1)  003502  001401                  BEQ     .+4             ;BRANCH IF OK
  (1)  003504  104000                  HLT                     ;FPS NOT EQUAL TO 47410
  (1)
 1026
```

```
1027  003506  174067  175308          STF     AC0,ANS1        ;STORE RESULT IN ANS1
1028  003512  173427  140500          CMPF    #-3,AC0         ;CHECK RESULT
1029  003516  170000                  CFCC                    ;GET CC
1030  003520  001401                  BEQ     .+4
1031  003522  104002                  HLT     2               ;0 = 9 NOT -37
1032
1033
1034                                  ;*****************************************************************
 (2)                                  ;TEST 16        TEST OF ADDD AND SUBD
 (2)                                  ;*****************************************************************
 (2)  003524  104400                  TST16:  SCOPE
1035
1036  003526  170127  047600          LDFPS   #47600          ;SET DOUBLE MODE
1037  003532  172527  141400          LDD     #-32.,,AC1      ;LOAD A -32.
1038  003536  172127  040640          ADDD    #5,AC1          ;ADD 5 TO -32.
1039  003542  170200                  STFPS   FPS             ;STORE FLOATING POINT STATUS
 (1)  003544  022700  047610          CMP     #47610,FPS      ;CHECK FLOATING POINT STATUS
 (1)  003550  001401                  BEQ     .+4             ;BRANCH IF OK
 (1)  003552  104000                  HLT                     ;FPS NOT EQUAL TO 47610
 (1)
1040
1041  003554  174177  000376          STD     AC1,@#ANS1      ;GET RESULT
1042  003560  173527  141330          CMPD    #-27,,AC1       ;CHECK RESULT
1043  003564  170000                  CFCC                    ;GET CC
1044  003566  001401                  BEQ     .+4
1045  003570  104004                  HLT     4               ;-32.+5 NOT -27
1046
1047  003572  172667  000362          LDD     DALTA,AC2       ;LOAD ALT 1'S
1048  003576  172267  000366          ADDD    DALTB,AC2       ;ADD OTHER 1'S
1049  003602  170200                  STFPS   FPS             ;STORE FLOATING POINT STATUS
 (1)  003604  022700  047600          CMP     #47600,FPS      ;CHECK FLOATING POINT STATUS
 (1)  003610  001401                  BEQ     .+4             ;BRANCH IF OK
 (1)  003612  104000                  HLT                     ;FPS NOT EQUAL TO 47600
 (1)
1050
1051  003614  174277  000336          STD     AC2,@#ANS1      ;GET RESULT
1052  003620  173627  040500          CMPD    #3,AC2          ;CHECK RESULT
1053  003624  170000                  CFCC                    ;GET CC
1054  003626  001401                  BEQ     .+4
1055  003630  104004                  HLT     4               ;ANS1 NOT DALTAN
1056
1057  003632  173267  000322          SUBD    DALTA,AC2       ;SUBTRACT IT BACK
1058  003636  170200                  STFPS   FPS             ;STORE FLOATING POINT STATUS
 (1)  003640  022700  047600          CMP     #47600,FPS      ;CHECK FLOATING POINT STATUS
 (1)  003644  001401                  BEQ     .+4             ;BRANCH IF OK
 (1)  003646  104000                  HLT                     ;FPS NOT EQUAL TO 47600
 (1)
1059
1060  003650  174267  175136          STD     AC2,ANS1        ;GET THE RESULT
1061  003654  173667  000320          CMPD    DALTC,AC2       ;CHECK IT
1062  003660  170000                  CFCC                    ;GET CC
1063  003662  001401                  BEQ     .+4
1064  003664  104004                  HLT     4               ;ANS1 NOT DALTA
```

```
 1066                                   ;*****************************************************************
  (2)                                   ;TEST 17        TEST OF MULF AND DIVF
  (2)                                   ;*****************************************************************
  (2)  003666  104400                   TST17:  SCOPE
 1067
 1068  003670  170127  047400                   LDFPS   #47400          ;LOAD FLOATING MODE
 1069  003674  172727  040740                   LDF     #7,AC3          ;LOAD A7
 1070  003700  171327  140500                   MULF    #-3,AC3         ;X -3
 1071  003704  170200                           STFPS   FPS             ;STORE FLOATING POINT STATUS
  (1)  003706  022700  047410                   CMP     #47410,FPS      ;CHECK FLOATING POINT STATUS
  (1)  003712  001401                           BEQ     .+4             ;BRANCH IF OK
  (1)  003714  104000                           HLT                     ;FPS NOT EQUAL TO 47410
  (1)
 1072
 1073  003716  174367  175070                   STF     AC3,ANS1        ;GET RESULT
 1074  003722  173727  141250                   CMPF    #-21.,AC3       ;CHECK RESULT
 1075  003726  170000                           CFCC                    ;GET CC
 1076  003730  001401                           BEQ     .+4
 1077  003732  104002                           HLT     2               ;X -3 NOT -21.
 1078
 1079  003734  174727  140740                   DIVF    #-7,AC3         ;DIVIDE BY -3
 1080  003740  170200                           STFPS   FPS             ;STORE FLOATING POINT STATUS
  (1)  003742  022700  047400                   CMP     #47400,FPS      ;CHECK FLOATING POINT STATUS
  (1)  003746  001401                           BEQ     .+4             ;BRANCH IF OK
  (1)  003750  104000                           HLT                     ;FPS NOT EQUAL TO 47400
  (1)
 1081
 1082  003752  174367  175034                   STF     AC3,ANS1        ;GET RESULT
 1083  003756  173727  040500                   CMPF    #3,AC3          ;CHECK RESULT
 1084  003762  170000                           CFCC                    ;GET CC
 1085  003764  001401                           BEQ     .+4
 1086  003766  104002                           HLT     .               ;-21. / -7 NOT 3
```

```
1088                                    ;*****************************************************************
 (2)                                    ;TEST 20        TEST OF MULD AND DIVD
 (2)                                    ;*****************************************************************
 (2)  003770  104400                    TST20:  SCOPE
1089
1090  003772  170127  047600                    LDFPS   #47600          ;LOAD DOUBLE MODE
1091  003776  172427  140640                    LDD     #-5.,AC0        ;LOAD A -5
1092  004002  171027  140500                    MULD    #-3.,AC0        ;MUL BY -3
1093  004006  170200                            STFPS   FPS             ;STORE FLOATING POINT STATUS
 (1)  004010  022700  047600                    CMP     #47600,FPS      ;CHECK FLOATING POINT STATUS
 (1)  004014  001401                            BEQ     .+4             ;BRANCH IF OK
 (1)  004016  104000                            HLT                     ;FPS NOT EQUAL TO 47600
 (1)
1094
1095  004020  174067  174766                    STD     AC0,ANS1        ;GET RESULT
1096  004024  173427  041160                    CMPD    #15.,,AC0       ;CHECK RESULT
1097  004030  170000                            CFCC                    ;GET CC
1098  004032  001401                            BEQ     .+4
1099  004034  104004                            HLT     4               ;-3 X -3 NOT +15.
1100
1101  004036  174427  140400                    DIVD    #-2,AC0         ;15. / -2
1102  004042  170200                            STFPS   FPS             ;STORE FLOATING POINT STATUS
 (1)  004044  022700  047610                    CMP     #47610,FPS      ;CHECK FLOATING POINT STATUS
 (1)  004050  001401                            BEQ     .+4             ;BRANCH IF OK
 (1)  004052  104000                            HLT                     ;FPS NOT EQUAL TO 47610
 (1)
1103
1104  004054  174067  174732                    STD     AC0,ANS1        ;STORE RESULT
1105  004060  173427  140760                    CMPD    #-7.5,AC0       ;CHECK RESULT
1106  004064  170000                            CFCC                    ;GET CC
1107  004066  001401                            BEQ     .+4
1108  004070  104004                            HLT     4               ;15. / -2 NOT -7.5
```

```
 1110                                   ;*****************************************************************
  (2)                                   ;TEST 21        TEST OF LDCFD,LDCDF
  (2)                                   ;*****************************************************************
  (2)  004072  104400           TST21:  SCOPE
 1111
 1112  004074  170127  047600           LDFPS   #47600          ;SET DOUBLE MODE
 1113  004100  172477  005024           LDD     @AD0101,AC0     ;LOAC A 0,-1,0,-1
 1114  004104  012700  011102           MOV     #D1010,FPS      ;GET ADDRESS OF DATA
 1115  004110  177420                    LDCFD   (FPS)+,AC0      ;LOAD A -1,0,0,0
 1116  004112  022700  011106           CMP     #D1010+4,FPS    ;INC BY 4?
 1117  004116  001401                    BEQ     .+4
 1118  004120  104000                    HLT                     ;FPS NOT #D1010+4
 1119
 1120  004122  170200                    STFPS   FPS             ;STORE FLOATING POINT STATUS
  (1)  004124  022700  047610           CMP     #47610,FPS      ;CHECK FLOATING POINT STATUS
  (1)  004130  001401                    BEQ     .+4             ;BRANCH IF OK
  (1)  004132  104000                    HLT                     ;FPS NOT EQUAL TO 47610
  (1)
 1121
 1122  004134  174067  174652           STD     AC0,ANS1        ;GET ANSWER
 1123  004140  173467  005106           CMPD    D1000,AC0       ;CHECK RESULT
 1124  004144  170000                    CFCC                    ;GET CC
 1125  004146  001401                    BEQ     .+4
 1126  004150  104004                    HLT     4               ;AC0 NOT -1,0,0,0
 1127
 1128  004152  172567  004726           LDD     D0101,AC1       ;LOAD A 0,-1,0,-1
 1129  004156  170001                    SETF                    ;SET FLOATING MODE
 1130  004160  012700  011112           MOV     #D1010+10,FPS   ;GET ADDRESS OF DATA +10
 1131  004164  177540                    LDCDF   -(FPS),AC1      ;LOAD A -1,0
 1132  004166  022700  011102           CMP     #D1010,FPS      ;ADDRESS DECREMENT BY 10?
 1133  004172  001401                    BEQ     .+4
 1134  004174  104000                    HLT                     ;FPS NOT #D1010
 1135
 1136  004176  170200                    STFPS   FPS             ;STORE FLOATING POINT STATUS
  (1)  004200  022700  047410           CMP     #47410,FPS      ;CHECK FLOATING POINT STATUS
  (1)  004204  001401                    BEQ     .+4             ;BRANCH IF OK
  (1)  004206  104000                    HLT                     ;FPS NOT EQUAL TO 47410
  (1)
 1137
 1138  004210  170011                    SETD
 1139  004212  174167  174574           STD     AC1,ANS1        ;GET RESULT
 1140  004216  173567  004676           CMPD    WEIRD,AC1       ;CHECK RESULT
 1141  004222  170000                    CFCC
 1142  004224  001401                    BEQ     .+4
 1143  004226  104004                    HLT     4               ;RESULT NOT -1,1,0,-1
 1144
 1145  004230  170127  047440           LDFPS   #47440          ;SET DOUBLE AND TRUNCATE MODES
 1146  004234  177567  004642           LDCDF   D1010,AC1       ;LOAD IT
 1147  004240  174167  174546           STD     AC1,ANS1        ;GET RESULT
 1148  004244  173567  004642           CMPD    D1001,AC1       ;CHECK RESULT
 1149  004250  170200                    STFPS   FPS
 1150  004252  170000                    CFCC                    ;GET CC
 1151  004254  001401                    BEQ     .+4
 1152  004256  104004                    HLT     4               ;AC1 NOT -1,0,0,1
```

```
 1153
 1154  004260  170127  047400                LDFPS   #47400          ;SET ROUND AND FLOATING MODES
 1155  004264  177567  004612                LDCDF   D1010,AC1       ;LOAD A -1,0
 1156  004270  174167  174516                STD     AC1,ANS1        ;GET THE RESULT
 1157  004274  170200                        STFPS   FPS
 1158  004276  022767  000001  174510        CMP     #1,ANS2         ;CHECK WORD 2 FOR A 1
 1159  004304  001401                        BEQ     .+4
 1160  004306  104004                        HLT     4               ;LDCDF DID NOT ROUND PROPERLY
 1161
 1162                                ;*****************************************************************
  (2)                                ;TEST 22        TEST OF STCFD,STCDF
  (2)                                ;*****************************************************************
  (2)  004310  104400                TST22:  SCOPE
 1163
 1164  004312  170127  040200                LDFPS   #40200          ;SET DOUBLE MODE
 1165  004316  172667  004560                LDD     D1010,AC2       ;LOAD A -1,0,-1,0
 1166  004322  170001                        SETF                    ;SET FLOATING MODE
 1167  004324  176202                        STCDF   AC2,AC2         ;CLEAR RIGHT HALF
 1168  004326  170011                        SETD                    ;SET DOUBLE MODE
 1169  004330  174267  174456                STD     AC2,ANS1        ;GET RESULT
 1170  004334  173667  004712                CMPD    D1000,AC2       ;IS IT -1,0,0,0
 1171  004340  170000                        CFCC                    ;GET CC
 1172  004342  001401                        BEQ     .+4
 1173  004344  104004                        HLT     4               ;AC1 NOT -1,0,0,0
 1174
 1175  004346  172767  004716                LDD     D0111,AC3       ;LOAD A 0,-1,0,-1
 1176  004352  176303                        STCFD   AC3,AC3         ;CLEAR OUT RIGHT HALF!?
 1177  004354  174367  174432                STD     AC3,ANS1        ;GET RESULT
 1178  004360  173767  004676                CMPD    D0100X,AC3      ;CHECK RESULT
 1179  004364  170000                        CFCC                    ;GET CC
 1180  004366  001401                        BEQ     .+4
 1181  004370  104004                        HLT     4               ;ANS3 NOT 100,0,0 (ROUND)
 1182
 1183  004372  172467  004672                LDD     D0111,AC0       ;LOAD 0,-1,0,-1
 1184  004376  170127  000040                LDFPS   #40             ;FLOATING AND TRUNCATE MODES
 1185  004402  176000                        STCFD   AC0,AC0         ;CLEAN RIGHT HALF
 1186  004404  170011                        SETD                    ;SET DOUBLE
 1187  004406  174067  174400                STD     AC0,ANS1        ;GET RESULT
 1188  004412  173467  004632                CMPD    D0100,AC0       ;CHECK IT
 1189  004416  170000                        CFCC                    ;GET CC
 1190  004420  001401                        BEQ     .+4
 1191  004422  104004                        HLT     4               ;AC0 NOT 0,-1,0,0 (TRUNCATE)
 1192
 1193                                ;*****************************************************************
  (2)                                ;TEST 23        TEST OF LDCIF,LDCID,STCFI,STCDI
  (2)                                ;*****************************************************************
  (2)  004424  104400                TST23:  SCOPE
 1194
 1195  004426  170127  047400                LDFPS   #47400          ;FLOATING MODE
 1196  004432  177027  000005                LDCIF   #5,AC0          ;STORE A 5
 1197  004436  170200                        STFPS   FPS             ;STORE FLOATING POINT STATUS
  (1)  004440  022700  047400                CMP     #47400,FPS      ;CHECK FLOATING POINT STATUS
  (1)  004444  001401                        BEQ     .+4             ;BRANCH IF OK
  (1)  004446  104000                        HLT                     ;FPS NOT EQUAL TO 47400
```

```
 (1)
1198
1199  004458  174067  174336          STF     AC0,ANS1        ;GET THE RESULT
1200  004454  173427  040040          CMPF    #5,AC0          ;CHECK IT
1201  004460  170000                  CFCC                    ;GET CC
1202  004462  001401                  BEQ     .+4
1203  004464  104002                  HLT     2               ;AC0 NOT 5.0
1204
1205  004466  175400                  STCFI   AC0,FPS         ;CONVERT IT BACK
1206  004470  022700  000005          CMP     #5,FPS          ;CHECK RESULT
1207  004474  001401                  BEQ     .+4
1208  004476  104000                  HLT                     ;FPS NOT 5
1209
1210  004500  170011                  SETD                    ;SET DOUBLE MODE
1211  004502  172567  004376          LDD     DB101,AC1       ;LOAD JUNK
1212  004506  012700  011102          MOV     #D1010,FPS      ;LOAD ADDRESS OF DATA
1213  004512  177120                  LDCID   (FPS)+,AC1      ;CONVERT TO -1.0
1214  004514  022700  011104          CMP     #D1010+2,FPS    ;CHECK ADDRESS
1215  004520  001401                  BEQ     .+4
1216  004522  104000                  HLT                     ;FPS NOT #D1010+2
1217
1218  004524  170200                  STFPS   FPS             ;STORE FLOATING POINT STATUS
 (1)  004526  022700  047610          CMP     #47610,FPS      ;CHECK FLOATING POINT STATUS
 (1)  004532  001401                  BEQ     .+4             ;BRANCH IF OK
 (1)  004534  104000                  HLT                     ;FPS NOT EQUAL TO 47610
 (1)
1219
1220  004536  174167  174250          STD     AC1,ANS1        ;GET RESULT
1221  004542  173527  140200          CMPD    #-1,AC1         ;CHECK RESULT
1222  004546  170000                  CFCC                    ;GET CC
1223  004550  001401                  BEQ     .+4
1224  004552  104004                  HLT     4               ;AC1 NOT -1.0
1225
1226  004554  175500                  STCDI   AC1,FPS         ;CONVERT IT BACK
1227  004556  022700  177777          CMP     #-1,FPS         ;CHECK RESULT
1228  004562  001401                  BEQ     .+4
1229  004564  104000                  HLT                     ;FPS NOT -1
1230
1231  004566  170001                  SETF                    ;SET FLOATING MODE
1232  004570  177227  054321          LDCIF   #54321,AC2      ;LOAD 54321
1233  004574  174267  174212          STF     AC2,ANS1        ;GET RESULT
1234  004600  170200                  STFPS   FPS             ;GET THE STATUS
1235  004602  173667  004500          CMPF    FSTO1,AC2       ;CHECK IT
1236  004606  170000                  CFCC                    ;CHECK CC
1237  004610  001401                  BEQ     .+4
1238  004612  104002                  HLT     2               ;AC2 NOT 54321.
1239
1240  004614  052737  000017  177776  BIS     #17,@#PS        ;SET PS
1241  004622  175667  174164          STCFI   AC2,ANS1        ;CONVERT IT BACK
1242  004626  013700  177776          MOV     @#PS,FPS        ;GET PS
1243  004632  042700  177760          BIC     #177760,FPS     ;CLEAR JUNK
1244  004636  005700                  TST     FPS             ;IS IT 0
1245  004640  001401                  BEQ     .+4             ;SKIP IF OK
1246  004642  104000                  HLT                     ;PS NOT 0
```

```
1247  004644  170200                        STFPS   FPS             ;GET THE FPS
1248
1249  004646  022767  054321  174136        CMP     #54321,ANS1     ;CHECK RESULT
1250  004654  001401                        BEQ     .+4
1251  004656  104001                        HLT     1               ;ANS1 NOT 54321
1252
1253                                ;*****************************************************************
 (2)                                ;TEST 24        TEST OF LDCLF,LDCLD,STCFL,STCDL
 (2)                                ;*****************************************************************
 (2)  004660  104400                TST24:  SCOPE
1254
1255  004662  170127  047500                LDFPS   #47500          ;FLOATING AND LONG MODES
1256  004666  177327  177773                LDCLF   #-5,AC3         ;LOAD A -5
1257  004672  000401                        BR      .+4
1258  004674  000000                        HALT                    ;LDCLF INCREMENTED BY 4 NOT 2
1259
1260  004676  170200                        STFPS   FPS             ;STORE FLOATING POINT STATUS
 (1)  004700  022700  047510                CMP     #47510,FPS      ;CHECK FLOATING POINT STATUS
 (1)  004704  001401                        BEQ     .+4             ;BRANCH IF OK
 (1)  004706  104000                        HLT                     ;FPS NOT EQUAL TO 47510
 (1)
1261
1262  004710  174367  174076                STF     AC3,ANS1        ;GET THE RESULT
1263  004714  173727                        CMPF    (7)+,AC3        ;CHECK IT
1264  004716  144640                        144640
1265  004720  170000                        CFCC                    ;GET CC
1266  004722  001401                        BEQ     .+4
1267  004724  104002                        HLT     2               ;AC3 NOT -5
1268
1269  004726  012700  011156                MOV     #AANS1,FPS      ;SET UP ADDRESS OF ADDRESS
1270  004732  175730                        STCFL   AC3,@(FPS)+     ;STORE IN ANS1
1271  004734  022700  011160                CMP     #AANS1+2,FPS    ;CHECK ADDRESS
1272  004740  001401                        BEQ     .+4
1273  004742  104000                        HLT                     ;ADDRESS IN FPU NOT AANS1+2
1274
1275  004744  170200                        STFPS   FPS             ;GET STATUS
1276  004746  022767  177773  174036        CMP     #-5,ANS1        ;CHECK LEFT HALF
1277  004754  001401                        BEQ     .+4
1278  004756  104002                        HLT     2               ;LEFT NOT -5
1279
1280  004760  005767  174030                TST     ANS2            ;CHECK RIGHT HALF OF RESULT
1281  004764  001401                        BEQ     .+4
1282  004766  104002                        HLT     2               ;ANS2 NOT 0
1283
1284  004770  170011                        SETD                    ;SET DOUBLE MODE
1285  004772  177067  004302                LDCLD   DST01,AC0       ;LOAD WEIRD NUMBER
1286  004776  170200                        STFPS   FPS             ;STORE FLOATING POINT STATUS
 (1)  005000  022700  047700                CMP     #47700,FPS      ;CHECK FLOATING POINT STATUS
 (1)  005004  001401                        BEQ     .+4             ;BRANCH IF OK
 (1)  005006  104000                        HLT                     ;FPS NOT EQUAL TO 47700
 (1)
1287
1288  005010  174067  173776                STD     AC0,ANS1        ;GET RESULT
1289  005014  173467  004266                CMPD    FST01,AC0       ;CHECK IT
```

```
 1290  000020  170000                        CFCC                    ;GET CC
 1291  000022  001401                        BEQ     .+4
 1292  000024  104004                        HLT     4               ;ACB NOT F9TO1
 1293
 1294  000026  012700  001016                MOV     #ANS3,FPS
 1295  000032  175440                        STCDL   AC0,-(FPS)      ;GET IT BACK
 1296  000034  022700  001012                CMP     #ANS1,FPS       ;CHECK ADDRESS
 1297  000040  001401                        BEQ     .+4
 1298  000042  104000                        HLT                     ;FPS NOT #ANS1
 1299
 1300  000044  170200                        STFPS   FPS             ;STORE FLOATING POINT STATUS
  (1)  000046  022700  047700                CMP     #47700,FPS      ;CHECK FLOATING POINT STATUS
  (1)  000052  001401                        BEQ     .+4             ;BRANCH IF OK
  (1)  000054  104000                        HLT                     ;FPS NOT EQUAL TO 47700
  (1)
 1301
 1302  000056  170200                        STFPS   FPS
 1303  000060  026767  004214  173724        CMP     D5TO1,ANS1      ;CHECK LEFT
 1304  000066  001401                        BEQ     .+4
 1305  000070  104002                        HLT     2               ;ANS1 NOT
 1306
 1307  000072  026767  004204  173714        CMP     D5TO1+2,ANS2    ;CHECK RIGHT
 1308  000100  001401                        BEQ     .+4
 1309  000102  104002                        HLT     2               ;ANS2 NOT
 1310
 1311                                ;*****************************************************************
  (2)                                ;TEST 25        TEST OF MODF AND MODD
  (2)                                ;*****************************************************************
  (2)  000104  104400                TST25:  SCOPE
 1312
 1313  000106  170127  047400                LDFPS   #47400          ;SET FLOATING MODE
 1314  000112  172467  004054                LDF     DALTB+2,AC0     ;LOAD 52525,52525 INTO AC0
 1315  000116  171427  040200                MODF    #1,AC0          ;MODF BY A 1 +MGT+
 1316  000122  170200                        STFPS   FPS             ;STORE FLOATING POINT STATUS
  (1)  000124  022700  047404                CMP     #47404,FPS      ;CHECK FLOATING POINT STATUS
  (1)  000130  001401                        BEQ     .+4             ;BRANCH IF OK
  (1)  000132  104000                        HLT                     ;FPS NOT EQUAL TO 47404
  (1)
 1317
 1318  000134  174067  173652                STF     AC0,ANS1        ;GET RESULT
 1319  000140  170300                        TSTF    AC0             ;CHECK FRACTION
 1320  000142  170000                        CFCC
 1321  000144  001401                        BEQ     .+4
 1322  000146  104002                        HLT     2               ;FRACTION NOT 0
 1323
 1324  000150  174167  173636                STF     AC1,ANS1        ;GET IT
 1325  000154  173567  004012                CMPF    DALTB+2,AC1     ;CHECK INTEGER
 1326  000160  170000                        CFCC
 1327  000162  001401                        BEQ     .+4
 1328  000164  104002                        HLT     2               ;INTEGER IS NOT 52525,52525
 1329
 1330  000166  172627  040200                LDF     #1,AC2          ;LOAD A 1
 1331  000172  171667  004042                MODF    D20,AC2         ;MOD BY 20000,0
 1332  000176  170200                        STFPS   FPS             ;STORE FLOATING POINT STATUS
```

```
   (1)  005200  022700  047400          CMP     #47400,FPS      ;CHECK FLOATING POINT STATUS
   (1)  005204  001401                  BEQ     .+4             ;BRANCH IF OK
   (1)  005206  104000                  HLT                     ;FPS NOT EQUAL TO 47400
   (1)
  1333
  1334  005210  174267  173576          STF     AC2,ANS1        ;GET IT
  1335  005214  173667  004020          CMPF    D20,AC2         ;CHECK FRACT
  1336  005220  170000                  CFCC
  1337  005222  001401                  BEQ     .+4
  1338  005224  104002                  HLT     2               ;RESULT NOT 20000,0
  1339
  1340  005226  174367  173560          STF     AC3,ANS1        ;GET INT
  1341  005232  170503                  TSTF    AC3             ;CHECK FOR 0
  1342  005234  170000                  CFCC
  1343  005236  001401                  BEQ     .+4
  1344  005240  104002                  HLT     2               ;RESULT NOT 0
  1345
  1346  005242  170011                  SETD                    ;SET DOUBLE MODE
  1347  005244  172467  003760          LDD     D46,AC0         ;LOAD A 40000,0,0,0
  1348  005250  171467  003744          MODD    D37,AC0         ;MOD BY 37400,0,0,0
  1349  005254  173467  003730          CMPD    D40,AC0         ;CHECK FOR 40000,0,0,0
  1350  005260  170000                  CFCC
  1351  005262  001401                  BEQ     .+4
  1352  005264  104004                  HLT     4               ;RESULT NOT 40000,0,0,0
  1353
  1354  005266  174367  173520          STD     AC3,ANS1        ;GET THE RESULT
  1355  005272  170503                  TSTD    AC3             ;CHECK FOR 0
  1356  005274  170000                  CFCC
  1357  005276  001401                  BEQ     .+4
  1358  005300  104004                  HLT     4               ;RESULT NOT 0
  1359
  1363                          ;*****************************************************************
   (2)                          ;TEST 26        LDD OF -0
   (2)                          ;*****************************************************************
   (2)  005302  104400          TST26:  SCOPE
  1364
  1365  005304  170127  047600          LDFPS   #47600          ;DOUBLE MODE
  1366  005310  172467  003674          LDD     D48,AC0         ;LOAD DUMMY DATA
  1367  005314  172467  003626  151:    LDD     DMZERO,AC0      ;LOAD A -0
  1368  005320  170200                  STFPS   FPS             ;STORE FLOATING POINT STATUS
   (1)  005322  170367  173564          STST    FEC             ;STORE EXCEPTION CODES
   (1)  005326  022700  147614          CMP     #147614,FPS     ;CHECK FLOATING POINT STATUS
   (1)  005332  001401                  BEQ     .+4             ;BRANCH IF OK
   (1)  005334  104000                  HLT                     ;FPS NOT EQUAL TO 147614
   (1)
   (1)  005336  022767  000014  173466  CMP     #14,FEC         ;CHECK FLOATING EXCEPTION CODE
   (1)  005344  001401                  BEQ     .+4             ;BRANCH IF OK
   (1)  005346  104000                  HLT                     ;FEC NOT EQUAL TO 14
   (1)
   (1)  005350  022767  005314  173456  CMP     #5314,FEA       ;CHECK FLOATING EXCEPTION ADDRESS
   (1)  005356  001401                  BEQ     .+4             ;BRANCH IF OK
   (1)  005360  104000                  HLT                     ;FEA NOT EQUAL TO 5314
   (1)
  1369
```

```
 1370  005362  174067  173424          STD     AC0,ANS1        ;GET RESULT FOR TYPING
 1371  005366  173467  003616          CMPD    D40,AC0         ;DID IT CHANGE AC0?
 1372  005372  170200                  STFPS   FPS             ;GET FPS FOR TYPING
 1373  005374  170000                  CFCC                    ;GET CC
 1374  005376  001401                  BEQ     .+4             ;SKIP IF OK
 1375  005400  104004                  HLT     4               ;RESULT IS WRONG
 1376
 1377  005402  170127  040200          LDFPS   #40200          ;DOUBLE MODE
 1378  005406  172467  003534          LDF     ZERO,AC0        ;LOAD A -0
 1379  005412  170200                  STFPS   FPS             ;STORE FLOATING POINT STATUS
   (1) 005414  022767  040214          CMP     #40214,FPS      ;CHECK FLOATING POINT STATUS
   (1) 005420  001401                  BEQ     .+4             ;BRANCH IF OK
   (1) 005422  104000                  HLT                     ;FPS NOT EQUAL TO 40214
   (1)
 1380
 1381  005424  174067  173362          STD     AC0,ANS1        ;GET RESULT
 1382  005430  173467  003512          CMPD    ZERO,AC0        ;CMP TO -0
 1383  005434  170200                  STFPS   FPS             ;GET FPS
 1384  005436  170000                  CFCC                    ;GET CC
 1385  005440  001401                  BEQ     .+4             ;SKIP IF OK
 1386  005442  104004                  HLT     4               ;RESULT NOT -0
 1387
 1388                          ;*****************************************************************
   (2)                         ;TEST 27        MULF ERROR - OVERFLOW
   (2)                         ;*****************************************************************
   (2)  005444  104400          TST27:  SCOPE
 1389
 1390  005446  170127  001000          LDFPS   #1000           ;FLOATING/OVERFLOW
 1391  005452  013777  177776  005164  MOV     @PS,@FPVECT+2   ;SET THE MODE IN PS WORD
 1392  005460  005001                  CLR     R1              ;CLEAR FLAG WORD
 1393  005462  012777  011074  005152  MOV     #CHKERR,@FPVECT ;SET TRAP ADDRESS
 1394  005470  172427  076101          LDF     #1E36,AC0       ;LOAD A LARGE NUMBER INTO AC0,
 1395  005474  171027  076101      1$: MULF    #1E36,AC0       ;MULTIPLY BY A LARGE NUMBER
 1396  005500  174067  173306          STF     AC0,ANS1        ;GET FOR TYPING
 1397  005504  170200                  STFPS   FPS             ;GET STATUS
 1398  005506  005701                  TST     R1              ;DID IT TRAP?
 1399  005510  001001                  BNE     3$              ;YES
 1400  005512  104002                  HLT     2               ;DID NOT TRAP ON OVERFLOW
 1401  005514                      3$:
   (1) 005514  170200                  STFPS   FPS             ;STORE FLOATING POINT STATUS
   (1) 005516  170367  173310          STST    FEC             ;STORE EXCEPTION CODES
   (1) 005522  022700  101002          CMP     #101002,FPS     ;CHECK FLOATING POINT STATUS
   (1) 005526  001401                  BEQ     .+4             ;BRANCH IF OK
   (1) 005530  104000                  HLT                     ;FPS NOT EQUAL TO 101002
   (1)
   (1) 005532  022767  000010  173272  CMP     #10,FEC         ;CHECK FLOATING EXCEPTION CODE
   (1) 005540  001401                  BEQ     .+4             ;BRANCH IF OK
   (1) 005542  104000                  HLT                     ;FEC NOT EQUAL TO 10
   (1)
   (1) 005544  022767  005474  173262  CMP     #5474,FEA       ;CHECK FLOATING EXCEPTION ADDRESS
   (1) 005552  001401                  BEQ     .+4             ;BRANCH IF OK
   (1) 005554  104000                  HLT                     ;FEA NOT EQUAL TO 5474
   (1)
 1402
```

```
1483                                    ;*****************************************************************
 (2)                                    ;TEST 38        DIVF ERROR - UNDERFLOW
 (2)                                    ;*****************************************************************
 (2)  005556  104400                    TST38:  SCOPE
1484
1485  005560  170127  002000                    LDFPS   #2000           ;FLOATING/UNDERFLOW
1486  005564  005001                    CLR     R1              ;CLEAR FLAG WORD
1487  005566  172427  002252            LDF     #1E-36,AC0      ;LOAD A SMALL NUMBER
1488  005572  174427  076101            1$:     DIVF    #1E36,AC0       ;DIVIVE BY A LARGE NUMBER
1489  005576  174067  173210            STF     AC0,ANS1        ;GET FOR TYPING
1410  005602  170200                    STFPS   FPS             ;GET STATUS
1411  005604  005701                    TST     R1              ;DID IT TRAP?
1412  005606  001001                    BNE     3$              ;SKIP IF SET
1413  005610  104002                    HLT     2               ;DID NOT TRAP ON UNDERFLOW
1414  005612                    3$:
 (1)  005612  170200                    STFPS   FPS             ;STORE FLOATING POINT STATUS
 (1)  005614  170367  173212            STST    FEC             ;STORE EXCEPTION CODES
 (1)  005620  022700  102000            CMP     #102000,FPS     ;CHECK FLOATING POINT STATUS
 (1)  005624  001401                    BEQ     .+4             ;BRANCH IF OK
 (1)  005626  104000                    HLT                     ;FPS NOT EQUAL TO 102000
 (1)
 (1)  005630  022767  000012  173174    CMP     #12.,FEC        ;CHECK FLOATING EXCEPTION CODE
 (1)  005636  001401                    BEQ     .+4             ;BRANCH IF OK
 (1)  005640  104000                    HLT                     ;FEC NOT EQUAL TO 12
 (1)
 (1)  005642  022767  005572  173164    CMP     #5572,FEA       ;CHECK FLOATING EXCEPTION ADDRESS
 (1)  005650  001401                    BEQ     .+4             ;BRANCH IF OK
 (1)  005652  104000                    HLT                     ;FEA NOT EQUAL TO 5572
 (1)
1415
1416                                    ;*****************************************************************
 (2)                                    ;TEST 31        STCFI ERROR - CONVERSION(6)
 (2)                                    ;*****************************************************************
 (2)  005654  104400                    TST31:  SCOPE
1417
1418  005656  170127  000400                    LDFPS   #400            ;FLOATING/INTEGER/CONVERSION
1419  005662  005001                    CLR     R1              ;CLEAR FLAG WORD
1420  005664  172527  076101            LDF     #1E36,AC1       ;LOAD LARGE NUMBER
1421  005670  175567  173116            1$:     STCFI   AC1,ANS1        ;TRY TO STUFF INTO 16 BITS
1422  005674  170200                    STFPS   FPS             ;GET STATUS
1423  005676  005701                    TST     R1              ;TRAP FLAG SET?
1424  005700  001001                    BNE     3$              ;SKIP IF SET
1425  005702  104000                    HLT                     ;DID NOT TRAP ON CONVERT
1426  005704                    3$:
 (1)  005704  170200                    STFPS   FPS             ;STORE FLOATING POINT STATUS
 (1)  005706  170367  173120            STST    FEC             ;STORE EXCEPTION CODES
 (1)  005712  022700  100405            CMP     #100405,FPS     ;CHECK FLOATING POINT STATUS
 (1)  005716  001401                    BEQ     .+4             ;BRANCH IF OK
 (1)  005720  104000                    HLT                     ;FPS NOT EQUAL TO 100405
 (1)
 (1)  005722  022767  000006  173102    CMP     #6,FEC          ;CHECK FLOATING EXCEPTION CODE
 (1)  005730  001401                    BEQ     .+4             ;BRANCH IF OK
 (1)  005732  104000                    HLT                     ;FEC NOT EQUAL TO 6
 (1)
```

```
   (1)  000734  022767  000670  173872          CMP     #5678,FEA       ;CHECK FLOATING EXCEPTION ADDRESS
   (1)  000742  001401                          BEQ     .+4             ;BRANCH IF OK
   (1)  000744  104000                          HLT                     ;FEA NOT EQUAL TO 5678
   (1)
  1427
  1428                                  ;*****************************************************************
   (2)                                  ;TEST 32        DIVF BY 0 ERROR
   (2)                                  ;*****************************************************************
   (2)  000746  104400                  TST32:  SCOPE
  1429
  1430  000750  170127  000000                  LDFPS   #0              ;FLOATING
  1431  000754  005001                          CLR     R1              ;CLEAR FLAG
  1432  000756  174527  000000          1$:     DIVF    #0,AC1          ;DIVIDE BY 0
  1433  000762  170200                          STFPS   FPS             ;GET STATUS
  1434  000764  005701                          TST     R1              ;CHECK FLAG
  1435  000766  001001                          BNE     3$              ;SKIP IF SET
  1436  000770  104000                          HLT                     ;DIVIDE BY 0 DID NOT TRAP
  1437  000772                          3$:
   (1)  000772  170200                          STFPS   FPS             ;STORE FLOATING POINT STATUS
   (1)  000774  170367  173832                  STST    FEC             ;STORE EXCEPTION CODES
   (1)  001000  022700  100000                  CMP     #100000,FPS     ;CHECK FLOATING POINT STATUS
   (1)  001004  001401                          BEQ     .+4             ;BRANCH IF OK
   (1)  001006  104000                          HLT                     ;FPS NOT EQUAL TO 100000
   (1)
   (1)  001010  022767  000004  173814          CMP     #4,FEC          ;CHECK FLOATING EXCEPTION CODE
   (1)  001016  001401                          BEQ     .+4             ;BRANCH IF OK
   (1)  001020  104000                          HLT                     ;FEC NOT EQUAL TO 4
   (1)
   (1)  001022  022767  005756  173804          CMP     #5756,FEA       ;CHECK FLOATING EXCEPTION ADDRESS
   (1)  001030  001401                          BEQ     .+4             ;BRANCH IF OK
   (1)  001032  104000                          HLT                     ;FEA NOT EQUAL TO 5756
   (1)
  1438
  1439                                  ;*****************************************************************
   (2)                                  ;TEST 33        LDF -0 ERROR
   (2)                                  ;*****************************************************************
   (2)  001034  104400                  TST33:  SCOPE
  1440
  1441  001036  170127  004000                  LDFPS   #4000           ;FLOATING/UNDEFINED VARIABLE
  1442  001042  005001                          CLR     R1              ;CLEAR FLAG
  1443  001044  172667  003076          1$:     LDF     ZMEERO,AC2      ;LOAD AN UNDEFINED VARIABLE
  1444  001050  170200                          STFPS   FPS             ;GET STATUS
  1445  001052  005701                          TST     R1              ;CHECK FLAG
  1446  001054  001001                          BNE     3$              ;SKIP IF SET
  1447  001056  104000                          HLT                     ;LOAD OF -0 DID NOT TRAP
  1448  001060                          3$:
   (1)  001060  170200                          STFPS   FPS             ;STORE FLOATING POINT STATUS
   (1)  001062  170367  172744                  STST    FEC             ;STORE EXCEPTION CODES
   (1)  001066  022700  104014                  CMP     #104014,FPS     ;CHECK FLOATING POINT STATUS
   (1)  001072  001401                          BEQ     .+4             ;BRANCH IF OK
   (1)  001074  104000                          HLT                     ;FPS NOT EQUAL TO 104014
   (1)
   (1)  001076  022767  000014  172726          CMP     #14,FEC         ;CHECK FLOATING EXCEPTION CODE
   (1)  001104  001401                          BEQ     .+4             ;BRANCH IF OK
```

```
(1)  000186  104000                    HLT                     ;FEC NOT EQUAL TO 14
(1)
(1)  000110  022767  000044  172716     CMP     #0044,FEA       ;CHECK FLOATING EXCEPTION ADDRESS
(1)  000116  001401                    BEQ     .+4             ;BRANCH IF OK
(1)  000120  104000                    HLT                     ;FEA NOT EQUAL TO 0044
(1)
1449
1450                            ;*****************************************************************
(2)                             ;TEST 34        OPCODE ERROR
(2)                             ;*****************************************************************
(2)  000122  104400             TST34:  SCOPE
1451
1452  000124  170127  000000            LDFPS   #0              ;FLOATING
1453  000130  005001                    CLR     R1              ;CLEAR FLAG
1454  000132  177777             1$:    177777                  ;ILLEGAL OPCODE
1455  000134  170200                    STFPS   FPS             ;GET STATUS
1456  000136  005701                    TST     R1              ;CHECK FLAG
1457  000140  001001                    BNE     3$              ;SKIP IF SET
1458  000142  104000                    HLT                     ;NOT AN ILLEGAL OPCODE?
1459  000144                     3$:
(1)   000144  170200                    STFPS   FPS             ;STORE FLOATING POINT STATUS
(1)   000146  170367  172660            STST    FEC             ;STORE EXCEPTION CODES
(1)   000152  022700  100000            CMP     #100000,FPS     ;CHECK FLOATING POINT STATUS
(1)   000156  001401                    BEQ     .+4             ;BRANCH IF OK
(1)   000160  104000                    HLT                     ;FPS NOT EQUAL TO 100000
(1)
(1)   000162  022767  000002  172642    CMP     #2,FEC          ;CHECK FLOATING EXCEPTION CODE
(1)   000170  001401                    BEQ     .+4             ;BRANCH IF OK
(1)   000172  104000                    HLT                     ;FEC NOT EQUAL TO 2
(1)
(1)   000174  022767  000132  172632    CMP     #0132,FEA       ;CHECK FLOATING EXCEPTION ADDRESS
(1)   000202  001401                    BEQ     .+4             ;BRANCH IF OK
(1)   000204  104000                    HLT                     ;FEA NOT EQUAL TO 0132
(1)
1460
1461                            ;*****************************************************************
(2)                             ;TEST 35        UBREAK TRAP
(2)                             ;*****************************************************************
(2)  000206  104400             TST35:  SCOPE
1462
1463  000210  005001                    CLR     R1              ;CLEAR FLAG
1464  000212  013767  177776  000100    MOV     @#PS,SAVSTS     ;GET PS
1465  000220  005037  177776            CLR     @#PS            ;ZERO IT
1466  000224  012703  000314            MOV     #314,R3         ;LOAD 314 INTO
1467  000230  170003                    LDUB                    ;UBREAK REG,
1468  000232  170127  000020            LDFPS   #20             ;FLOATING/MAINT
1469  000236  174427  040200     1$:    DIVF    #1,AC0          ;DIVIDE SHOULD TRAP
1470  000242  170200                    STFPS   FPS             ;GET STATUS
1471  000244  005701                    TST     R1              ;CHECK FLAG
1472  000246  001001                    BNE     3$              ;SKIP IF NOT SET
1473  000250  104000                    HLT                     ;UBREAK DID NOT TRAP
1474  000252                     3$:
(1)   000252  170200                    STFPS   FPS             ;STORE FLOATING POINT STATUS
(1)   000254  170367  172552            STST    FEC             ;STORE EXCEPTION CODES
```

```
    (1)  000260  022700  100020          CMP     #100020,FPS     ;CHECK FLOATING POINT STATUS
    (1)  000264  001401                  BEQ     .+4             ;BRANCH IF OK
    (1)  000266  104000                  HLT                     ;FPS NOT EQUAL TO 100020
    (1)
    (1)  000270  022767  000016  172534  CMP     #16,FEC         ;CHECK FLOATING EXCEPTION CODE
    (1)  000276  001401                  BEQ     .+4             ;BRANCH IF OK
    (1)  000300  104000                  HLT                     ;FEC NOT EQUAL TO 16
    (1)
    (1)  000302  022767  000236  172524  CMP     #0236,FEA       ;CHECK FLOATING EXCEPTION ADDRESS
    (1)  000310  001401                  BEQ     .+4             ;BRANCH IF OK
    (1)  000312  104000                  HLT                     ;FEA NOT EQUAL TO 0236
    (1)
 1475    000314  170127  040000          LDFPS   #40000          ;CLEAR FPS
 1476    000320  016737  002774  177776  MOV     SAVSTS,@#PS     ;RESTORE PS
 1477
 1478                                    ;*****************************************************************
  (2)                                    ;TEST 36        ADDF ERROR - OVERFLOW
  (2)                                    ;*****************************************************************
  (2)    000326  104400                  TST36:  SCOPE
 1479
 1480    000330  170127  001000          LDFPS   #1000           ;FLOATING/OVERFLOW
 1481    000334  005001                  CLR     R1              ;CLEAR FLAG
 1482    000336  172767  002574          LDF     DBIG,AC3        ;LOAD A BIG NUMBER
 1483    000342  172367  002570  181:     ADDF    DBIG,AC3        ;MAKE OVERFLOW
 1484    000346  170200                  STFPS   FPS             ;GET STATUS
 1485    000350  174367  172436          STF     AC3,ANS1        ;GET RESULT
 1486    000354  005701                  TST     R1              ;FLAG SET?
 1487    000356  001001                  BNE     3$              ;SKIP IF SET
 1488    000360  104002                  HLT     2               ;DID NOT OVERFLOW
 1489    000362                  3$:
    (1)  000362  170200                  STFPS   FPS             ;STORE FLOATING POINT STATUS
    (1)  000364  170367  172442          STST    FEC             ;STORE EXCEPTION CODES
    (1)  000370  022700  101006          CMP     #101006,FPS     ;CHECK FLOATING POINT STATUS
    (1)  000374  001401                  BEQ     .+4             ;BRANCH IF OK
    (1)  000376  104000                  HLT                     ;FPS NOT EQUAL TO 101006
    (1)
    (1)  000400  022767  000010  172424  CMP     #10,FEC         ;CHECK FLOATING EXCEPTION CODE
    (1)  000406  001401                  BEQ     .+4             ;BRANCH IF OK
    (1)  000410  104000                  HLT                     ;FEC NOT EQUAL TO 10
    (1)
    (1)  000412  022767  000342  172414  CMP     #0342,FEA       ;CHECK FLOATING EXCEPTION ADDRESS
    (1)  000420  001401                  BEQ     .+4             ;BRANCH IF OK
    (1)  000422  104000                  HLT                     ;FEA NOT EQUAL TO 0342
    (1)
 1490
 1491                                    ;*****************************************************************
  (2)                                    ;TEST 37        SUBF ERROR - UNDERFLOW
  (2)                                    ;*****************************************************************
  (2)    000424  104400                  TST37:  SCOPE
 1492
 1493    000426  170127  002000          LDFPS   #2000           ;FLOATING/UNDERFLOW
 1494    000432  005001                  CLR     R1              ;CLEAR FLAG
 1495    000434  172427  000430          LDF     #.87E-37,AC0    ;LOAD SMALL NUMBER
 1496    000440  173027  000504          SUBF    #.89E-37,AC0    ;SUB BIG NUMBER
```

```
1497  006444  174067  172342          STF     AC0,ANS1        ;GET RESULT
1498  006450  170200                  STFPS   FPS             ;GET STATUS
1499  006452  005701                  TST     R1              ;FLAG SET?
1500  006454  001001                  BNE     3$              ;SKIP IF SET
1501  006456  104002                  HLT     2               ;NO UNDERFLOW
1502  006460                  3$:
 (1)  006460  170200                  STFPS   FPS             ;STORE FLOATING POINT STATUS
 (1)  006462  170367  172344          STST    FEC             ;STORE EXCEPTION CODES
 (1)  006466  022700  102014          CMP     #102014,FPS     ;CHECK FLOATING POINT STATUS
 (1)  006472  001401                  BEQ     .+4             ;BRANCH IF OK
 (1)  006474  104000                  HLT                     ;FPS NOT EQUAL TO 102014
 (1)
 (1)  006476  022767  000012  172326  CMP     #12,FEC         ;CHECK FLOATING EXCEPTION CODE
 (1)  006504  001401                  BEQ     .+4             ;BRANCH IF OK
 (1)  006506  104000                  HLT                     ;FEC NOT EQUAL TO 12
 (1)
 (1)  006510  022767  006440  172316  CMP     #6440,FEA       ;CHECK FLOATING EXCEPTION ADDRESS
 (1)  006516  001401                  BEQ     .+4             ;BRANCH IF OK
 (1)  006520  104000                  HLT                     ;FEA NOT EQUAL TO 6440
 (1)
1503
1504  006522  104400                  SCOPE
1505
1506  006524  170127  040000          LDFPS   #40000          ;I/O FPS
1507  006530  012777  000760  004104  MOV     #FLTERR,@FPVECT ;RESTORE VECTOR
1508  006536  012777  000340  004100  MOV     #340,@FPVECT+2  ;RESTORE LEVEL 7
1509  006544  004767  001230          JSR     PC,CODE         ;DO FLOATING POINT EXERCISER THROUGHOUT MEMORY
1510  006550  032737  001000  177570  BIT     #SW09,@#SWR     ;RELOCATE CODE?
1511  006556  001402                  BEQ     RELEX           ;RELOCATE
1512  006560  000167  002540          JMP     DONE+2          ;DO NOT RELOCATE
```

```
1514                                        ;RELOCATOR AND EXECUTOR
1515  000564  012737  000672  000004  RELEX: MOV     #TRAP4,@#4
1516  000572  012701  010000              MOV     #CODE,R1        ;LOAD STARTING POINT
1517  000576  012704  020000              MOV     #20000,R4       ;LOAD DESTINATION
1518  000602  004767  000102              JSR     PC,MOVE         ;MOVE THE CODE THERE
1519  000606  004737  020000              JSR     PC,@#20000      ;EXECUTE THE 5K BANK
1520  000612  012704  024000              MOV     #24000,R4
1521  000616  010401                1$:   MOV     R4,R1           ;GET THE ADDRESS
1522  000620  010146                      MOV     R1,-(6)         ;SAVE ON THE STACK
1523  000622  012704  020000              MOV     #20000,R4       ;SET TO GET THE LOADER
1524  000626  004767  000056              JSR     PC,MOVE         ;GET THE PAGE
1525  000632  011604                      MOV     (6),R4          ;GET THE DESTINATION ADDRESS
1526  000634  012701  010000              MOV     #CODE,R1        ;DATA TO BE TRANSFERED
1527  000640  004767  000044              JSR     PC,MOVE         ;MOVE IT
1528  000644  004776  000000              JSR     PC,0(6)         ;EXECUTE IT
1529  000650  012604                      MOV     (6)+,R4         ;GET DEST ADDRESS
1530  000652  012701  020000              MOV     #20000,R1       ;GET SAVE ADDRESS
1531  000656  004767  000026              JSR     PC,MOVE         ;RESTORE LOADER
1532  000662  022704  160000              CMP     #160000,R4      ;END OF WORLD
1533  000666  001353                      BNE     1$              ;NO - LOOP
1534  000670  000402                      BR      X2X             ;YES - SKIP STACK CLEANER
1535
1536  000672  062706  000010        TRAP4: ADD     #10,SP          ;CLEAR THE STACK
1537  000676  012737  000006  000004 X2X:  MOV     #6,@#4          ;RESET 4
1538  000704  000167  002412              JMP     DONE
1539
1540                                        ;MOVE SUBROUTINE - MOVES 1K CHUNK FROM (1) TO (4)
1541
1542  000710  010246                MOVE:  MOV     R2,-(6)         ;SAVE R2
1543  000712  012702  002000              MOV     #2000,R2        ;LOAD A 1K COUNT
1544  000716  016746  171054              MOV     PS,-(6)         ;SAVE PSW
1545  000722  005046                      CLR     -(6)            ;PUT 0 ON STACK
1546  000724  012746  000732              MOV     #1$,-(6)        ;PUT RETURN ADD ON STACK
1547  000730  000002                      RTI                     ;CONT. WITH NEW PS
1548  000732  011114                1$:   MOV     (1),(4)         ;MOVE A WORD
1549  000734  010400                      MOV     R4,FPS          ;SAVE NEW ADD FOR TYPE OUT
1550  000736  022124                      CMP     (1)+,(4)+       ;DID WD GET TO NEW LOCATION OK
1551  000740  001401                      BEQ     .+4             ;YES
1552  000742  104000                      HLT                     ;DATA TRANSFER BAD R1=OLD ADD +2 R4=NEW ADD +2
1553  000744  005302                      DEC     R2              ;DEC COUNT
1554  000746  001371                      BNE     1$              ;LOOP UNTIL DONE
1555  000750  012746  000756              MOV     #2$,-(6)        ;STOR RETURN ADD
1556  000754  000002                      RTI                     ;GET OLD PSW AND NEW RETURN ADD
1557  000756  012602                2$:   MOV     (6)+,R2         ;RESTORE R2
1558  000760  000207                      RTS     PC              ;RETURN
```

```
                    010000                  .=10000
1561  010000  012737  000037  001000  CODE1:  MOV     #37,@#ICNT
1562
 (2)                                        ;*****************************************************************
 (2)                                        ;TEST 40        TEST FOR CONVERSION, ADDD AND SUBD
 (2)                                        ;*****************************************************************
 (2)  010006  104400                        TST40:  SCOPE
1563  010010  170127  047600                        LDFPS   #47600
1564  010014  177327  000004                        LDCID   #4,AC3          ;LOAD AC3 WITH A FLOATING 4
1565  010020  177227  000002                        LDCID   #2,AC2          ;LOAD AC2 WITH A FLOATING 2
1566  010024  172302                        ADDD    AC2,AC3         ;ADD 2+4 = 6 IN AC3
1567  010026  173302                        SUBD    AC2,AC3         ;SUB 2 FROM 6
1568  010030  175737  001012                        STCDI   AC3,@#ANS1      ;@#ANS1 SHOULD =4
1569  010034  022737  000004  001012                CMP     #4,@#ANS1               ;DOES @#ANS1 =4
1570  010042  170200                        STFPS   FPS
1571  010044  001401                        BEQ     .+4             ;YES
1572  010046  104001                        HLT     1               ;@#ANS1 SHOULD = 4
1573
1574                                        ;*****************************************************************
 (2)                                        ;TEST 41        LDD AND STD TEST
 (2)                                        ;*****************************************************************
 (2)  010050  104400                        TST41:  SCOPE
1575  010052  177027  000005                        LDCID   #5,AC0          ;LOAD AC0 WITH A FLOATING 5
1576  010056  174005                        STD     AC0,AC5         ;NOW PUT IT INTO AC5
1577  010060  172605                        LDD     AC5,AC2         ;NOW PUT IT INTO AC2
1578  010062  172200                        ADDD    AC0,AC2         ;ADD 5 TO 5
1579  010064  173205                        SUBD    AC5,AC2         ;SUB 5 FROM 10
1580  010066  175637  001012                        STCDI   AC2,@#ANS1      ;PUT ANS INTO @#ANS1
1581  010072  022737  000005  001012                CMP     #5,@#ANS1               ;WERE THE TWO AC'S EQUAL
1582  010100  170200                        STFPS   FPS
1583  010102  001401                        BEQ     .+4             ;YES
1584  010104  104001                        HLT     1               ;ANS1 SHOULD = 5
1585
1586                                        ;*****************************************************************
 (2)                                        ;TEST 42        MULD AND DIVD TEST
 (2)                                        ;*****************************************************************
 (2)  010106  104400                        TST42:  SCOPE
1587  010110  177227  000052                        LDCID   #52,AC2         ;LOAD AC2 WITH FLOATING DOUBLE 52
1588  010114  174204                        STD     AC2,AC4         ;PUT IT INTO AC4
1589  010116  177327  000025                        LDCID   #25,AC3         ;LOAD AC3 WITH FL DB 25
1590  010122  171304                        MULD    AC4,AC3         ;MUL 52X25 RESULT IN AC3
1591  010124  174704                        DIVD    AC4,AC3         ;DIV 52 INTO RESULT IN AC3
1592  010126  171304                        MULD    AC4,AC3         ;MUL 52X25
1593  010130  172304                        ADDD    AC4,AC3         ;ADD AC4 TO AC3 TO MAKE 53 TIMES
1594  010132  012704  000025                        MOV     #25,R4          ;SET UP COUNTER
1595  010136  173304                SUBT:   SUBD    AC4,AC3         ;SUB 52 FROM AC3 25 TIMES
1596  010140  005304                        DEC     R4
1597  010142  001375                        BNE     SUBT            ;SUB 53 TIMES
1598  010144  175737  001012                        STCDI   AC3,@#ANS1      ;ANS SHOULD BE 52
1599  010150  170200                        STFPS   FPS
1600  010152  022737  000052  001012                CMP     #52,@#ANS1      ;IS ANS CORRECT?
1601  010160  001401                        BEQ     .+4             ;YES
1602  010162  104001                        HLT     1               ;ANS SHULD BE 52
```

```
1624                                    ;*****************************************************************
 (2)                                    ;TEST 43        EXERCISER TEST
 (2)                                    ;*****************************************************************
 (2)  010164  104400            TST43:  SCOPE
1625  010166  177027  025252            LDCID   #25252,AC0      ;LOAD 25252 INTO AC0
1626  010172  177127  000025            LDCID   #25,AC1         ;LOAD AC1 WITH 25
1627  010176  171100                    MULD    AC0,AC1         ;MUL 25252X25 ANS IN AC1
1628  010200  174137  001012            STD     AC1,@#ANS1      ;SAV ANS IN @#ANS1
1629  010204  012702  000024            MOV     #24,R2          ;SET UP COUNT
1630  010210  174005                    STD     AC0,AC5         ;PUT 25252 INTO AC5
1631  010212  172005            AAD:    ADDD    AC5,AC0         ;ADD 25252 TO 25252
1632  010214  005302                    DEC     R2              ;DO 25 TIMES
1633  010216  001375                    BNE     AAD             ;DONE?
1634  010220  174037  001022            STD     AC0,@#ANS5      ;LOAD TO PRINT
1635  010224  170200                    STFPS   FPS
1636  010226  173437  001012            CMPD    @#ANS1,AC0      ;IS ANS CORRECT?
1637  010232  170000                    CFCC
1638  010234  001401                    BEQ     .+4             ;YES
1639  010236  104010                    HLT     8.              ;EITHER THE ADD OR THE MUL DID NOT WORK
1640  010240  177327  000002            LDCID   #2,AC3          ;LOAD AC3 WITH A 2
1641  010244  012702  000013            MOV     #13,R2          ;SET UP COUNTER
1642  010250  174304                    STD     AC3,AC4         ;LOAD AC4 WITH A 2
1643  010252  171304            MMUL:   MULD    AC4,AC3         ;MUL 2 X 2
1644  010254  005302                    DEC     R2              ;DO 16 TIMES
1645  010256  001375                    BNE     MMUL
1646  010260  175703                    STCDI   AC3,R3          ;PUT ANS IN R3
1647  010262  010337  001012            MOV     R3,@#ANS1       ;NOW PUT IT INTO @#ANS1
1648  010266  022703  010000            CMP     #10000,R3       ;ANS SHOULD BE 10000
1649  010272  170200                    STFPS   FPS
1650  010274  001401                    BEQ     .+4             ;CONT, IF ANS IS CORRECT
1651  010276  104001                    HLT     1               ;ANS SHOULD BE 10000
1652  010300  177227  010000            LDCID   #10000,AC2      ;LOAD AC2 WITH 10000
1653  010304  177127  000002            LDCID   #2,AC1          ;LOAD AC1 WITH A 2
1654  010310  012702  000013            MOV     #13,R2          ;SET UP COUNTER
1655  010314  174601            DDIV:   DIVD    AC1,AC2         ;DIVD 2 INTO 65536 16 TIMES
1656  010316  005302                    DEC     R2              ;COUNT NO OF TIMES
1657  010320  001375                    BNE     DDIV            ;ARE WE DONE?
1658  010322  175603                    STCDI   AC2,R3          ;STOR ANS INTO R3
1659  010324  010237  001012            MOV     R2,@#ANS1       ;PUT IT INTO @#ANS1 FOR TYPING
1660  010330  022703  000002            CMP     #2,R3           ;IS ANS CORRECT
1661  010334  170200                    STFPS   FPS
1662  010336  001401                    BEQ     .+4             ;ANS IS CORRECT
1663  010340  104001                    HLT     1               ;ANS SHOULD EQUAL =2
```

```
 1645                                   ;*****************************************************************
  (2)                                   ;TEST 44        MODE ONE TEST
  (2)                                   ;*****************************************************************
  (2)  010342  104400           TST44:  SCOPE
 1646  010344  170127  047400           LDFPS   #47400
 1647  010350  177327  000005           LDCIF   #5,AC3          ;AC3=5
 1648  010354  012702  001012           MOV     #ANS1,R2        ;R2=ANS1
 1649  010360  177227  000007           LDCIF   #7,AC2          ;LOAD 7 INTO AC2
 1650  010364  174312                   STF     AC3,(R2)        ;ANS1 SHOULD =5=R2
 1651  010366  173737  001012           CMPF    @#ANS1,AC3      ;DOES @#ANS1 = 5
 1652  010372  170200                   STFPS   FPS
 1653  010374  170000                   CFCC
 1654  010376  001401                   BEQ     .+4             ;YES BRANCH
 1655  010400  104001                   HLT     1               ;ANS1 SHOULD CONTAIN 5
 1656  010402  171212                   MULF    (R2),AC2        ;MUL 5 X 7,AC2 = 35
 1657  010404  012703  000006           MOV     #6,R3           ;SET UP COUNTER
 1658  010410  173212           SUBFM:  SUBF    (R2),AC2        ;SUB 5 FROM 35
 1659  010412  005303                   DEC     R3              ;DO 7 TIMES
 1660  010414  001375                   BNE     SUBFM
 1661  010416  175637  001012           STCFI   AC2,@#ANS1      ;@#ANS1 SHOULD =5
 1662  010422  022737  000005  001012   CMP     #5,@#ANS1       ;DOES @#ANS1 =5
 1663  010430  170200                   STFPS   FPS
 1664  010432  001401                   BEQ     .+4             ;BRANCH IF YES
 1665  010434  104001                   HLT     1               ;ANS SHOULD =5
 1666
 1667                                   ;*****************************************************************
  (2)                                   ;TEST 45        ADDD EXERCISER
  (2)                                   ;*****************************************************************
  (2)  010436  104400           TST45:  SCOPE
 1668  010440  170127  047600           LDFPS   #47600
 1669  010444  177027  000252           LDCID   #252,AC0        ;LOAD AC0 WITH 252
 1670  010450  177227  052525           LDCID   #52525,AC2      ;LOAD AC2 WITH 52525
 1671  010454  174204                   STD     AC2,AC4         ;AC4=52525
 1672  010456  171002                   MULD    AC2,AC0         ;AC0=52525 X 252
 1673  010460  012702  000251           MOV     #251,R2         ;SET UP COUNTER
 1674  010464  172204           AADDM:  ADDD    AC4,AC2         ;ADD 52525 TO 252
 1675  010466  005302                   DEC     R2              ;DO 252 TIMES
 1676  010470  001375                   BNE     AADDM           ;DONE?
 1677  010472  174237  001012           STD     AC2,@#ANS1      ;GET FOR PRINTING
 1678  010476  170200                   STFPS   FPS
 1679  010500  173402                   CMPD    AC2,AC0         ;DOES AC2 = AC0?
 1680  010502  170000                   CFCC
 1681  010504  001401                   BEQ     .+4             ;BRANCH IF EQUAL
 1682  010506  104004                   HLT     4               ;ANS SHOULD BE.....
```

```
1684
 (2)
 (2)
 (2)  010510  104400                    ;*****************************************************************
1685  010512  170127  047600            ;TEST 46        TEST DIVDE BY 0
1686  010516  177027  077777            ;*****************************************************************
1687  010522  177127  000000  TST46:  SCOPE
1688  010526  174401                    LDFPS   #47600
1689  010530  175437  001012            LDCID   #77777,AC0
1690  010534  170200                    LDCID   #0,AC1
1691  010536  170337  001032            DIVD    AC1,AC0         ;DIVIDE 0 INTO 77777
1692  010542  022737  077777  001012    STCDI   AC0,@#ANS1      ;LOAD @#ANS1 WITH 77777
1693  010550  001401                    STFPS   FPS
1694  010552  104001                    STST    @#FEC           ;GET @#FEC STATUS
1695  010554  022737  000004  001032    CMP     #77777,@#ANS1
1696  010562  001401                    BEQ     .+4
1697  010564  104000                    HLT     1               ;ANS SHOULD =77777
1698                                    CMP     #4,@#FEC                ;DID WE TRY TO DIV BY 0?
 (2)                                    BEQ     .+4             ;YES
 (2)                                    HLT                     ;FEC SHOULD =4
 (2)  010566  104400                    ;*****************************************************************
1699  010570  170127  047400            ;TEST 47        EXERCISER FOR ADDD,SUBD,MULD AND DIVD
1700  010574  170011                    ;*****************************************************************
1701  010576  177027  077777  TST47:  SCOPE
1702  010602  177127  002525            LDFPS   #47400
1703  010606  012702  000012            SETD                    ;SET DOUBLE MODE
1704  010612  174401                    LDCID   #77777,AC0      ;LOAD AC0 WITH 77777
1705  010614  171001                    LDCID   #2525,AC1       ;LOAD AC1 WITH 2525
1706  010616  172001                    MOV     #12,R2          ;SET UP COUNTER
1707  010620  173001            EXLOP:  DIVD    AC1,AC0         ;DIVDE 2525 INTO 77777
1708  010622  000302                    MULD    AC1,AC0         ;MUL 2525 X ANS
1709  010624  001372                    ADDD    AC1,AC0         ;ADD 2525 TO ANS
1710  010626  175437  001012            SUBD    AC1,AC0         ;SUB 2525 FROM ANS
1711  010632  170200                    DEC     R2              ;DO 12 TIMES
1712  010634  022737  077777  001012    BNE     EXLOP           ;DONE?
1713  010642  001401                    STCDI   AC0,@#ANS1      ;LOAD ANS INTO @#ANS1
1714  010644  104001                    STFPS   FPS
1715  010646  172437  011102    EX4:    CMP     #77777,@#ANS1   ;IS @#ANS1 CORRECT
1716  010652  012702  001012            BEQ     .+4             ;BRANCH IF CORRECT
1717  010656  174022                    HLT     1
1718  010660  172537  011102            LDD     @#D1010,AC0     ;GET DATA
1719  010664  174122                    MOV     #ANS1,R2
1720  010666  172737  001022            STD     AC0,(R2)+
1721  010672  170200                    LDD     @#D1010,AC1
1722  010674  173737  001012            STD     AC1,(R2)+
1723  010700  170000                    LDD     @#ANS5,AC3
1724  010702  001401                    STFPS   FPS
1725  010704  104010                    CMPD    @#ANS1,AC3
1726  010706  172537  011104            CFCC
1727  010712  174142                    BEQ     .+4
1728  010714  172012                    HLT     0,              ;ANS1 AND ANS5 SHOULD =1010
1729  010716  173042                    LDD     @#D0101,AC1
1730  010720  173001                    STD     AC1,-(R2)       ;STORED AC1 IN ANS5-8
1731  010722  175437  001012            ADDD    (R2),AC0
                                        SUBD    -(R2),AC0
                                        SUBD    AC1,AC0
                                        STCDI   AC0,@#ANS1
```

```
1732  010726  022737  000000  001012          CMP     #0,@#ANS1
1733  010734  170200                          STFPS   FPS
1734  010736  001401                          BEQ     .+4
1735  010740  104001                          HLT     1
1736  010742  170001                  MORE:   SETF                    ;SET FLOATING MODE
1737  010744  177027  000525                  LDCIF   #525,AC0        ;LOAD AC0 WITH 525
1738  010750  177127  000252                  LDCIF   #252,AC1        ;LOAD AC1 WITH 252
1739  010754  174104                          STF     AC1,AC4         ;LOAD AC4 WITH 252
1740  010756  172104                          ADDF    AC4,AC1         ;ADD 252 TO 252
1741  010760  172701                          LDF     AC1,AC3         ;PUT ANS IN AC3=524
1742  010762  173003                          SUBF    AC3,AC0         ;SUB 524 FROM 525
1743  010764  175037  001012                  STEXP   AC0,@#ANS1      ;1 IN ANS1
1744  010770  170200                          STFPS   FPS
1745  010772  022737  000001  001012          CMP     #1,@#ANS1       ;CORRECT  ANS SHOULD BE 1
1746  011000  001401                          BEQ     .+4
1747  011002  104001                          HLT     1               ;ANS SHOULD BE 5
1748  011004  177027  000021                  LDCIF   #21,AC0         ;LOAD AC0 WITH 21
1749  011010  171000                          MULF    AC0,AC0         ;21 TIMES 21 AC0 = 441
1750  011012  174427  040400                  DIVF    #2,AC0          ;DIV BY 2
1751  011016  171427  040200                  MODF    #1,,AC0         ;MUL 1 TIMES 441
1752  011022  170200                          STFPS   FPS             ;STOR FPS
1753  011024  175537  001012                  STCFI   AC1,@#ANS1      ;PUT ANS IN ANS1
1754  011030  022737  000220  001012          CMP     #220,@#ANS1     ;IS IT EQUAL?
1755  011036  001401                          BEQ     .+4             ;YES
1756  011040  104001                          HLT     1               ;SHOULD = 220
1757  011042  171427  041040                  MODF    #10.,AC0        ;GET FRACTION
1758  011046  175537  001012                  STCFI   AC1,@#ANS1      ;PUT AC0 INTO ANS1
1759  011052  022737  000005  001012          CMP     #5,@#ANS1       ;ANS1 SHOULD = 5
1760  011060  001401                          BEQ     .+4
1761  011062  104001                          HLT     1               ;ANS1 SHOULD = 5
1762  011064  104400                          SCOPE
1763  011066  105037  001001                  CLRB    @#ICNT+1
1764  011072  000207                          RTS     PC
```

```
1766  011074  012701  177777          CHKERR: MOV     #-1,R1          ;SET FLAG
1767  011100  000002                          RTI                     ;RETURN
1768
1769  011102  177777                  D1010:  -1
1770  011104  000000  177777  000000  D0101:  0,-1,0
1771  011112  177777  000000  000000  D1001:  -1,0,0
1772  011120                          DSMALL:
1773  011120  177777  000001  000000  WEIRD:  -1,1,0,-1
      011126  177777
1774  011130  011104                  AD0101: D0101
1775  011132  011112                  AD1001: D1001
1776  011134  011252                  AD1000: D1000
1777  011136  077777  000000  177777  DBIG:   77777,0,-1,0
      011144  000000
1778  011146  100000  000000  000000  DMBERO: 100000,0,0,0
      011154  000000
1779  011156  001012                  AANS1:  ANS1
1780  011160  040252  125252  125252  DALTA:  40252,125252,125252,125252
      011166  125252
1781  011170  040325  052525  052525  DALTB:  40325,52525,52525,52525
      011176  052525
1782  011200  040325  052525  052525  DALTC:  40325,52525,52525,52526
      011206  052526
1783  011210  040000  000000  000000  D40:    40000,0,0,0
      011216  000000
1784  011220  037400  000000  000000  D37:    37400,0,0,0
      011226  000000
1785  011230  040600  000000  000000  D46:    40600,0,0,0
      011236  000000
1786  011240  020000  000000  000000  D20:    20000,0,0,0
      011246  000000
1787  011250  000000                  D0100:  0
1788  011252  177777  000000  000000  D1000:  -1,0,0,0
      011260  000000
1789  011262  000100  000000  000000  D0100X: 100,0,0
1790  011270  000000  177777  177777  D0111:  0,-1,-1,-1
      011276  177777
1791  011300  000000  054321          D5T01:  0,54321
1792  011304  011300                  AD5T01: D5T01
1793  011306  043661  121000  000000  F5T01:  43661,121000,0,0
      011314  000000
1794  011316  000000                  SAVADR: 0
1795  011320  000000                  SAVST:  0
```

```
1826  011322  104400                DONE:   SCOPE
1827  011324  032737  002000  177570        BIT     #SW10,@#SWR     ;RING THE BELL?  *SW10*
1828  011332  001002                        BNE     1$              ;NO!
1829  011334  000004  000007                TYPE    ,BELL
1830
1831  011340  005046                1$:     CLR     -(6)            ;CLEAR TRACE TRAP
1832  011342  032737  010000  177570        BIT     #SW12,@#SWR     ;RUN WITH TRT?  *SW12*
1833  011350  001010                        BNE     2$
1834  011352  005167  001256                COM     TRPB
1835  011356  100005                        BPL     2$
1836  011360  052716  000020                BIS     #20,(6)         ;SET TRACE TRAP
1837  011364  012746  001100                MOV     #BEGIN,-(6)     ;JUMP TO START OF TEST
1838  011370  000002                        RTI
1839  011372  012746  001100        2$:     MOV     #BEGIN,-(6)     ;JUMP TO START OF TEST
1840  011376  013700  000042                MOV     @#42,R0         ;RERUN?
1841  011402  001405                        BEQ     RERUN
1842  011404  000005                        RESET
1843  011406  004710                LOGICAL: JSR    7,(0)           ;JMP TO MONITOR
1844  011410  000240                        NOP
1845  011412  000240                        NOP
1846  011414  000240                        NOP
1847  011416  000002                RERUN:  RTI
1848
1849  011420  032737  000400  177570  ,EMT:  BIT     #400,@#SWR      ;KILL LOOP OR LOOP ON SPEC. TEST  *SW8*
1850  011426  001404                        BEQ     1$
1851  011430  126767  166134  167342        CMPB    SWR,ICNT        ;ON RIGHT TEST?   *SW7-0*
1852  011436  001437                        BEQ     OVER
1853  011440  116703  166124        1$:     MOVB    SWR,R3          ;GET UB BITS
1854  011444  170003                        LDUB
1855  011446  032737  040000  177570        BIT     #SW14,@#SWR     ;LOOP ON TEST   *SW14*
1856  011454  001026                        BNE     KIT
1857  011456  032737  004000  177570        BIT     #SW11,@#SWR     ;KILL ITERATIONS   *SW11*
1858  011464  001012                        BNE     SAVLAD
1859  011466  105767  167307                TSTB    ICNT+1
1860  011472  001404                        BEQ     2$              ;BRANCH IF FIRST
1861  011474  126767  001136  167277        CMPB    TIMES,ICNT+1    ;DONE?
1862  011502  001013                        BNE     KIT             ;BRANCH IF NOT
1863  011504  112767  000001  167267  2$:   MOVB    #1,ICNT+1       ;FIRST ITERATION
1864  011512  105267  167262        SAVLAD: INCB    ICNT            ;COUNT TEST NUMBERS
1865  011516  011667  167266                MOV     (6),LAD         ;SAVE LOOP ADDRESS
1866  011522  016737  167252  177570        MOV     ICNT,@#SWR      ;DISPLAY TEST NO. AND ITERATION COUNT
1867  011530  000002                        RTI                     ;RETURN
1868
1869  011532  105267  167243        KIT:    INCB    ICNT+1
1870  011536  016737  167236  177570  OVER: MOV     ICNT,@#SWR      ;SET UP DISPLAY
1871  011544  005767  167240                TST     LAD             ;FIRST ONE?
1872  011550  001760                        BEQ     SAVLAD
1873  011552  016716  167232                MOV     LAD,(6)         ;FUDGE RETURN ADDRESS
1874  011556  000002                        RTI                     ;FIXES PS
```

```
1876  011560  032737  002000  177570  ,TRP1   BIT     @SW10,@#SWR     ;BELL ON ERROR?
1877  011566  001402                          BEQ     1$              ;NO - SKIP
1878  011570  000004  000007                  TYPE    ,BELL           ;RING BELL
1879  011574  004767  000444          1$:     JSR     PC,ERROR        ;COUNT THE NUMBER OF ERRORS
1880  011600  016446                          MOV     R4,-(6)
1881  011602  032737  020000  177570          BIT     @SW13,@#SWR     ;SKIP TYPEOUT IF SET
1882  011610  001074                          BNE     4$
1883  011612  000004  011616                  TYPE    ,.+2            ;,ASCII <15><12>
1884  011622  016646  000002                  MOV     2(6),-(6)       ;PUT ADDRESS OF INSTRUCTION ON STACK
1885  011626  162716  000002                  SUB     #2,(6)
1886  011632  011605                          MOV     (6),TTY         ;TYPE (6) IN OCTAL
 (1)  011634  004767  000222                  JSR     PC,PRINTR       ;TYPE LEADING ZERO'S
1887  011640  000004  012617                  TYPE    ,SPACE+3
1888  011644  010005                          MOV     R0,TTY          ;TYPE R0 IN OCTAL
 (1)  011646  004767  000210                  JSR     PC,PRINTR       ;TYPE LEADING ZERO'S
1889  011652  000004  012620                  TYPE    ,SPACE+4
1890  011656  012703  001012                  MOV     #ANS1,R3        ;ADDRESS OF DATA
1891  011662  113604                          MOVB    @(6)+,R4        ;AMOUNT OF DATA IN TABLE
1892  011664  001426                          BEQ     3$
1893  011666  100016                          BPL     2$              ;TYPE STACK?
1894  011670  016667  000006  167114          MOV     6(6),ANS1
1895  011676  016667  000010  167110          MOV     10(6),ANS2
1896  011704  016667  000012  167104          MOV     12(6),ANS3
1897  011712  016667  000014  167100          MOV     14(6),ANS4
1898  011720  042704  177600                  BIC     #177600,R4      ;CLEAR SIGN
1899  011724  000004  012620          2$:     TYPE    ,SPACE+4
1900  011730  012305                          MOV     (3)+,TTY        ;TYPE (3)+ IN OCTAL
 (1)  011732  004767  000124                  JSR     PC,PRINTR       ;TYPE LEADING ZERO'S
1901  011736  005304                          DEC     R4
1902  011740  001371                          BNE     2$
1903  011742  005700                  3$:     TST     FPS
1904  011744  100016                          BPL     4$
1905  011746  000004  012614                  TYPE    ,SPACE
1906  011752  170337  001032                  STST    @#FEC
1907  011756  013705  001032                  MOV     @#FEC,TTY       ;TYPE @#FEC IN OCTAL
 (1)  011762  004767  000074                  JSR     PC,PRINTR       ;TYPE LEADING ZERO'S
1908  011766  000004  012617                  TYPE    ,SPACE+3
1909  011772  016705  167036                  MOV     FEA,TTY         ;TYPE FEA IN OCTAL
 (1)  011776  004767  000060                  JSR     PC,PRINTR       ;TYPE LEADING ZERO'S
1910  012002  012604                  4$:     MOV     (6)+,R4
1911  012004  005737  177570                  TST     @#SWR           ;HALT ON ERROR
1912  012010  100001                          BPL     .+4             ;SKIP IF CONTINUE
1913  012012  000000                          HALT                    ;HALT ON ERROR!
1914  012014  032737  001000  177570          BIT     @#SW09,@#SWR    ;CHECK FOR INHIBIT LOOP ON ERROR
1915  012022  001001                          BNE     .+4             ;SKIP IF LOOP ON ERROR
1916  012024  000002                          RTI
1917  012026  105007  166747                  CLRB    ICNT+1
1918  012032  032737  000400  177570          BIT     @#SW08,@#SWR    ;CHECK FOR LOAD MICROBREAK
1919  012040  001234                          BNE     KIT             ;BRANCH IF NOT
1920  012042  113703  177570                  MOVB    @#SWR,R3        ;PUT MICROBREAK ADDRESS IN R3
1921  012046  170003                          LDUB                    ;LOAD MICROBREAK
1922  012050  000630                          BR      KIT             ;LOOP ON TEST UNTIL NO ERRORS
```

```
 1924                                   ;       SOCTAL          OCTAL TYPEOUT ROUTINE
  (1)                                   ;
  (1)                                   ;THIS ROUTINE IS USED TO TYPE AN OCTAL NUMBER ON THE TTY. IT WILL TYPE
  (1)                                   ;ALL 6 CHARACTERS, SUPPRESS LEADING ZEROES, TYPE AN 18 BIT ADDRESS, OR TYPE
  (1)                                   ;THE 16 BITS. IT IS CALLED VIA THE DUMP, SDUMP, DUMP18, OR BITYPE MACRO'S.
  (1)                                   ;
  (1)  012052  012767  170101  000140   BITYPS: MOV     #170101,.PR     ;SET BIT FLAG AND 16. CHARACTER COUNT
  (1)  012060  000411                           BR      .PTIT           ;NOW TYPE IT IN BIT FORM
  (1)  012062  112767  000001  000130   PRINTR: MOVB    #1,.PR          ;SET ZERO FILL SWITCH
  (1)  012070  000402                           BR      .+6             ;SKIP
  (1)  012072  005067  000122           PRINTS: CLR     .PR             ;SUPRESS LEADING ZERO'S
  (1)  012076  112767  177772  000115           MOVB    #-6.,.PR+1      ;SET COUNT
  (1)  012104  010446                   .PTIT:  MOV     R4,-(6)         ;SAVE R4
  (1)  012106  012704  012222                   MOV     #.PR+2,R4       ;SET POINTER TO FIRST ASCII CHAR.
  (1)  012112  105014                           CLRB    (4)             ;CLEAR FIRST BYTE
  (1)  012114  000411                           BR      .PRF            ;ROTATE FIRST BIT
  (1)  012116  105014                   .PRL:   CLRB    (4)             ;CLEAR BYTE OF CHARACTER
  (1)  012120  032767  000100  000072           BIT     #100,.PR        ;BIT TYPING MODE?
  (1)  012126  001004                           BNE     .PRF            ;YES - SKIP 2 ROTATES
  (1)  012130  006105                           ROL     TTY             ;ROTATE BIT INTO C
  (1)  012132  106114                           ROLB    (4)             ;PACK IT
  (1)  012134  006105                           ROL     TTY             ;ROTATE BIT INTO C
  (1)  012136  106114                           ROLB    (4)             ;PACK IT
  (1)  012140  006105                   .PRF:   ROL     TTY             ;ROTATE BIT INTO C
  (1)  012142  106114                           ROLB    (4)             ;PACK IT
  (1)  012144  105714                           TSTB    (4)             ;IS IT ZERO?
  (1)  012146  001402                           BEQ     .+6             ;SKIP INC
  (1)  012150  105267  000044                   INCB    .PR             ;SET FILL SWITCH
  (1)  012154  105767  000040                   TSTB    .PR             ;CHECK FILL SWITCH
  (1)  012160  001402                           BEQ     .+6             ;SKIP BITSET
  (1)  012162  152724  000060                   BISB    #'0,(4)+        ;MAKE INTO ASCII CHAR
  (1)  012166  105267  000027                   INCB    .PR+1           ;INC COUNT
  (1)  012172  001351                           BNE     .PRL            ;REPEAT
  (1)  012174  022704  012222                   CMP     #.PR+2,R4       ;EMPTY BUFFER?
  (1)  012200  001002                           BNE     .+6             ;SKIP IF NOT
  (1)  012202  112724  000060                   MOVB    #'0,(4)+        ;LOAD 1 ZERO
  (1)  012206  105014                           CLRB    (4)             ;NULL TERMINATOR
  (1)  012210  000004  012222                   TYPE    ,.PR+2          ;TYPE IT
  (1)  012214  012604                           MOV     (6)+,R4         ;RESTORE R4
  (1)  012216  000207                           RTS     PC              ;RETURN
  (1)  012220  000012                   .PR:    .BLKW   12              ;COUNT, SWITCH, AND OUTPUT BUFFER
 1925
 1926  012244  005267  166532           ERROR:  INC     ERRORS          ;COUNT ERRORS
 1927  012250  132737  000001  000041           BITB    #1,@#41         ;AUTO MODE?
 1928  012256  001412                           BEQ     1$              ;NO!
 1929  012260  022767  000010  166514           CMP     #10.,ERRORS     ;TOO MANY?
 1930  012266  001006                           BNE     1$              ;NOT YET
 1931  012270  013700  000042                   MOV     @#42,R0         ;GET ADDRESS
 1932  012274  001403                           BEQ     1$              ;FORGET IT IF ZERO
 1933  012276  005037  000042                   CLR     @#42            ;TRAP 42
 1934  012302  004710                           JSR     PC,(0)          ;CALL THE MONITOR
 1935  012304  000207                   1$:     RTS     PC              ;RETURN
```

```
1940  012306  012777  012582  000314  POWDWN: MOV     #ILLUP,@UPVEC   ;SET FOR FAST UP
1941  012314  012777  000340  000310          MOV     #340,@UPVEC+2   ;PR7:0
1942  012322  170246                          STFPS   -(6)            ;GET THE FPS
1943  012324  170011                          SETD
1944  012326  174046                          STD     AC0,-(6)        ;SAVE AC'S
1945  012330  174146                          STD     AC1,-(6)
1946  012332  174246                          STD     AC2,-(6)
1947  012334  174346                          STD     AC3,-(6)
1948  012336  172404                          LDD     AC4,AC0
1949  012340  174046                          STD     AC0,-(6)
1950  012342  172405                          LDD     AC5,AC0
1951  012344  174046                          STD     AC0,-(6)
1952  012346  010046                          MOV     R0,-(0)         ;SAVE REGISTERS
1953  012350  010146                          MOV     R1,-(0)
1954  012352  010246                          MOV     R2,-(0)
1955  012354  010346                          MOV     R3,-(0)
1956  012356  010446                          MOV     R4,-(0)
1957  012360  010546                          MOV     R5,-(0)
1958  012362  010667  000234                  MOV     SP,SAVE6        ;SAVE SP
1959  012366  012777  012376  000234          MOV     #POWUP,@UPVEC   ;SET UP VECTOR
1960  012374  000000                          HALT
1961
1962  012376  016706  000220          POWUP:  MOV     SAVE6,SP        ;GET SP
1963  012402  005001                          CLR     R1              ;WAIT LOOP FOR THE TTY
1964  012404  005201                  1$:     INC     R1
1965  012406  001376                          BNE     1$
1966  012410  012605                          MOV     (6)+,R5         ;GET THE REGISTERS
1967  012412  012604                          MOV     (6)+,R4
1968  012414  012603                          MOV     (6)+,R3
1969  012416  012602                          MOV     (6)+,R2
1970  012420  012601                          MOV     (6)+,R1
1971  012422  012600                          MOV     (6)+,R0
1972  012424  170011                          SETD
1973  012426  172426                          LDD     (6)+,AC0        ;RESTORE THE AC'S
1974  012430  174005                          STD     AC0,AC5
1975  012432  172426                          LDD     (6)+,AC0
1976  012434  174004                          STD     AC0,AC4
1977  012436  172726                          LDD     (6)+,AC3
1978  012440  172626                          LDD     (6)+,AC2
1979  012442  172526                          LDD     (6)+,AC1
1980  012444  172426                          LDD     (6)+,AC0
1981  012446  170126                          LDFPS   (6)+            ;RESTORE FPS
1982  012450  012777  012306  000146          MOV     #POWDWN,@DWNVEC ;SET UP THE POWER DOWN VECTOR
1983  012456  012777  000340  000142          MOV     #340,@DWNVEC+2
1984  012464  000004  012470                  TYPE    ,,+2            ;,ASCIZ <15><12>"POWER"
1985  012500  000002                          RTI
1986
1987  012502  000000          ILLUP:  HALT                            ;THE POWER UP SEQUENCE WAS STARTED
1988  012504  000776                  BR      .-2                     ; BEFORE THE POWER DOWN WAS COMPLETE
```

```
1998                                    ;       STYPE           MESSAGE TYPEOUT ROUTINE
  (1)
  (1)                                   ;THIS ROUTINE IS USE TO TYPE ASCII MESSAGES ON THE TTY. THE
  (1)                                   ;CALL CAN BE IN ONE OF 3 FORMS: 1) "TYPE ,ADR" - TYPES THE
  (1)                                   ;MESSAGE STARTIN IN LOCATION "ADR!", 2) "TYPE ,CHAR" - TYPES
  (1)                                   ;THE ASCII "CHAR", AND 3) "PRINT <<15><12>"MESSAGE"> - TYPES
  (1)                                   ;THE MESSAGE WHICH IS INLINE ASCII.
  (1)
  (1) 012506 010546                     10TS:   MOV     TTY,-(6)        ;SAVE TTY
  (1) 012516 017605 000002                      MOV     @2(6),TTY       ;GET ADDRESS TO BE TYPED
  (1) 012514 032705 177400                      BIT     #177400,TTY     ;IS IT A TYPE#?
  (1) 012520 001004                             BNE     1$              ;NO
  (1) 012522 010567 000004                      MOV     TTY,.TYPE       ;GET THE CHARACTER
  (1) 012526 012705 012612                      MOV     #.TYPE,TTY      ;FUDGE THE ADDRESS
  (1) 012532 105715                     1$:     TSTB    (TTY)           ;TERMINATOR?
  (1) 012534 001406                             BEQ     2$              ;GET OUT IF SO
  (1) 012536 112537 177506                      MOVB    (TTY)+,@#177506 ;LOAD AND TYPE THE CHARACTER
  (1) 012542 105737 177564                      TSTB    @#177564        ;IS THE PRINTER READY
  (1) 012546 100375                             BPL     .-4             ;WAIT UNTIL IT IS
  (1) 012550 000770                             BR      1$              ;GET THE NEXT CHARACTER
  (1) 012552 017646 000002              2$:     MOV     @2(6),-(6)      ;GET ADDRESS TO BE TYPED
  (1) 012556 062766 000002 000004               ADD     #2,4(6)         ;ADD 2 TO THE ADDRESS
  (1) 012564 022666 000002                      CMP     (6)+,2(6)       ;IS IT ,+2?
  (1) 012570 001006                             BNE     3$              ;NO
  (1) 012572 062705 000002                      ADD     #2,TTY          ;ADD 2 TO THE ADDRESS
  (1) 012576 042705 000001                      BIC     #1,TTY          ;BACK UP TO AN EVEN BYTE
  (1) 012602 010566 000002                      MOV     TTY,2(6)        ;RESTORE ADDRESS
  (1) 012606 012605                     3$:     MOV     (6)+,TTY        ;RESTORE TTY
  (1) 012610 000002                             RTI                     ;RETURN
  (1) 012612 000000                     .TYPE:  0                       ;CHARACTER TYPE LOCATION
1991 012614 000015 020040 000040        SPACE:  .ASCIZ  <15><12>"  "
1992 012622 000000                      SAVE6:  0
1993 012624 000024 000026               DWNVEC: 24,26                   ;POWER DOWN VECTOR ADDRESS
1994 012630 000024 000026               UPVEC:  24,26                   ;POWER UP VECTOR ADDRESS
1996 012634 000000                      TRPB:   0
1997 012636 000377                      TIMES:  377                     ;ITERATION COUNT
2000 012640 172160                      FPTADR: 172160                  ;FLOATING POINT ADDRESS ON THE 11/20
2001 012642 000244 000246               FPVECT: 244,246                 ;FLOATING POINT VECTOR ADDRESS
2002
2016
2018                    000001                  .END
```

DCFPOD,P11 SYMBOL TABLE

```
AAD     010212          AADDM   010464          AANS1   011156          AC0   =%000000
AC1   =%000001          AC2   =%000002          AC3   =%000003          AC4   =%000004
AC5   =%000005          ADANS1  002136          AD0101  011130          AD1000  011134
AD1021  011132          ADST01  011304          ANS1    001012          ANS2    001014
ANS3    001016          ANS4    001020          ANS5    001022          ANS6    001024
ANS7    001026          ANS8    001030          BEGIN   001100          BELL  = 000007
BITYPS  012052          CMKERR  011074          CODE    010000          DALTA   011160
DALTB   011170          DALTC   011200          DBIG    011136          DDIV    010314
DMZERO  011146          DONE    011322          DSMALL  011120          DWNVEC  012624
D0100   011250          D0100X  011262          D0101   011104          D0111   011270
D1000   011252          D1001   011112          D1010   011102          D20     011240
D37     011220          D40     011210          D46     011230          DST01   011300
ERROR   012244          ERRORS  001002          EXLOP   010012          EX4     010646
FEA     001034          FEC     001032          FLTERR  000760          FPS   =%000000
FPTADR  012640          FPVECT  012642          FST01   011306          ICNT    001000
ILLUP   012502          IOTS    012506          KIT     011532          LAD     001010
LDSC  = 170004          LDUB  = 170003          LOGICA  011406          MMUL    010252
MORE    010742          MOVE    006710          MRS   = 170006          M1120   001072
N     = 000050          OVER    011536          PC    =%000007          PCNT    001004
POWDWN  012306          POWUP   012376          PRINTR  012062          PRINTS  012072
PS    = 177776          RELEX   006564          RERUN   011410          RETURN=%000001
R0    =%000000          R1    =%000001          R2    =%000002          R3    =%000003
R4    =%000004          R5    =%000005          SAVADR  011310          SAVE6   012622
SAVLAD  011512          SAVSTS  011320          SP    =%000006          SPACE   012614
START   001036          STAB  = 170005          STOB  = 170007          SUBFM   010410
SUBT    010136          SWR   = 177570          SW00  = 000400          SW09  = 001000
SW10  = 002000          SW11  = 004000          SW12  = 010000          SW13  = 020000
SW14  = 040000          TIMES   012636          TRAP4   006672          TRPB    012634
TST1    001172          TST10   002300          TST11   002424          TST12   002606
TST13   003004          TST14   003174          TST15   003422          TST16   003524
TST17   003666          TST2    001310          TST20   003770          TST21   004072
TST22   004310          TST23   004424          TST24   004660          TST25   005104
TST26   005302          TST27   005444          TST3    001414          TST30   005556
TST31   005654          TST32   005746          TST33   006034          TST34   006122
TST35   006206          TST36   006326          TST37   006424          TST4    001600
TST40   010006          TST41   010050          TST42   010100          TST43   010164
TST44   010342          TST45   010436          TST46   010510          TST47   010566
TST5    001722          TST6    002044          TST7    002224          TTY   =%000005
TYPE  = 000004          UPVEC   012630          WEIRD   011120          X2X     006676
.BIT  = 176777          .EMT    011420          .PR     012220          .PRF    012140
.PRL    012116          .PTIT   012104          .TRP    011562          .TYPE   012612
.     = 012646
```

ERRORS DETECTED: 0

@DCFPOD,DCFPOD=DCFPOD.P11
RUN-TIME: 13 19 0 SECONDS
CORE USED: 8K

94

A5