FP11

LDEXP

MD-11-DCFPK-B

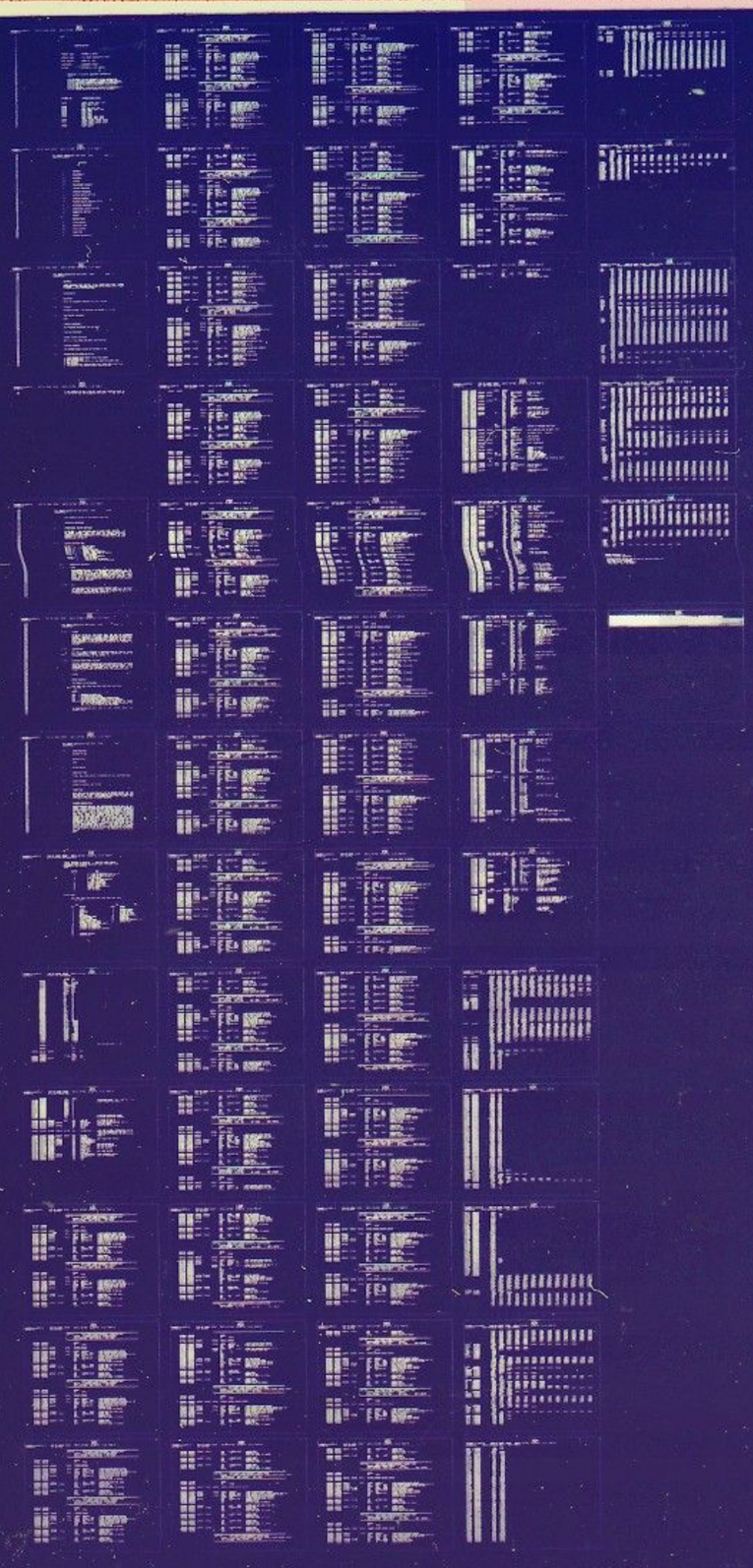
EP-DCFPK-B-DL-A

OCT 1976

FICHE 1 OF 1

digital

Made in U.S.A.



IDENTIFICATION

PRODUCT CODE: MAINDEC-11-DCFPA

PRODUCT NAME: F11 BASIC INSTRUCTION TESTS

DATE CREATED: MARCH 12, 1973

MAINTAINER: DIAGNOSTIC GROUP

AUTHORS: BOB BRAIN & KEN CHAPMAN

MAINDEC NO.	INSTRUCTIONS TESTED
DCFPA	LDFPS, STFPS, SETI, SETL SETF, SETD, CFCC
DCFPB	STST
DCFPC	LDF, LDD, STF, STD
DCFPD	ADDF, ADDD, SUBF, SUBD
DCFPE	CMPF, CMPD
DCFPF	MULF, MULD
DCFPG	DIVF, DIVD
DCFPH	CLRF, CLRD, TSTF, TSTD ABSF, ABSD, NEGF, NEGD
DCFPI	LDCFD, LDCDF, STCFD, STCDF
DCFPJ	LDCIF, LDCLF, LDCID, LDCLD STCFI, STCFL, STCDI, STCDL
DCFPK	LDEXP, STEXP
DCFPL	MODF, MODD

FP11 BASIC INSTRUCTION TEST DCFPA - DCFPL PAGE 2
TABLE OF CONTENTS

CONTENTS

FP11 BASIC INSTRUCTION TEST DCFPA - DCFPL PAGE 3
DESCRIPTION

1. ABSTRACT

THESE PROGRAMS TEST THE FP11 IN ALL MODES WITH FIXED NUMBER PATTERNS. THE PROGRAMS SHOULD BE RUN IN ORDER 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 - 17775

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.

```
;**
;**                    ?) THE DISPLAY ON THE 11/45 WILL SHOW THE ITERATION COUNT IN
;**                    "HE LEF" BYTE AND "ES" NUMBER IN "E RIGHT.  "O USE, SE" "E
```

FPJ: BASIC INSTRUCTION TEST DCFPA - DCFPL PAGE 4
DESCRIPTION

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. IF AN ERROR OCCURS, THAT TEST WILL BE LOOPED UPON UNTIL COMPLETION OF 256 CONSECUTIVE PASSES WITH NO ERRORS OF THE SUBTEST IF SW<9> SET TO A 1. 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 ..... LOOP ON ERROR
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 5.1.) IF A HLT IS EXECUTED. THE SUBTEST WILL BE LOOPED UPON UNTIL 256 CONSECUTIVE GOOD PASSES ARE COMPLETED IF SW<9> IS ON A 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 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)

THE FP11 INTERRUPT DISABLE BIT IS ALWAYS SET IN ALL OF THESE TESTS (EXCEPT DCFPA) SO 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.

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
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.

FP11 BASIC INSTRUCTION TEST DCFPA - DCFPL PAGE 6
DESCRIPTION

6.2 ERROR RECOVERY

RESTART AT 200

7. RESTRICTIONS

NONE

8. MISCELLANEOUS

8.1 EXECUTION TIME

A BELL WILL RING WITHIN 15 SECONDS WITH ALL SWITCHES DOWN.

8.2 STACK POINTER

STACK IS INITALLY SET TO 600

8.3 POWER FAIL

EACH 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

THESE PROGRAMS TEST ALL THE INSTRUCTIONS ON THE FP11 IN ALL MODES. EACH PROGRAM HAS MANY SUBTESTS (THE CODE BETWEEN 2 SCOPE STATEMENTS) WHICH ARE RUN 256 TIMES BEFORE CONTINUING TO THE NEXT. SW(11) ON A 1 CAUSES EACH SUBTEST TO BE RUN ONLY ONCE. SW(9) ON A 1 ENABLES LOOP ON ERROR. THE ADDRESS ICNT (LOC 1000) 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.

.END

```
.TITLE  MAINDEC-11-DCFPK-B      TEST OF LDEXP, STEXP
;COPYRIGHT 1972, DIGITAL EQUIPMENT CORP., MAYNARD, MASS
;PROGRAM BY KEN CHAPMAN
.REM*
```

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

OUTPUT FORM:

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

FPS

BIT	REASON
0	CARRY
1	OVERFLOW
2	ZERO
3	NEGATIVE
4	MAINTAINANCE MODE
5	TRUNCATE MODE
6	LONG INTEGER MODE
7	DOUBLE PRECISION MODE
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*

FEC

CODE	ERROR
0	ADDRESS ERROR
2	OPCODE ERROR
4	DIVIDE BY ZERO
6	CONVERSION ERROR
10	OVERFLOW
12	UNDERFLOW
14	UNDEFINED VARIABLE (-C)
16	UBREAK TRAP

```
                                .ENABL  ABS
000001                          N=      1
177776                          PS=     177776
177570                          SWR=    177570
177570                          DISPLAY=SWR
104400                          SCOPE=  TRAP
104000                          HLT=    EMT
000004                          TYPE=   IOT
000207                          BELL=   207
000000                          FPS=    %0
000000                          R0=     %0
000001                          R1=     %1
000002                          R2=     %2
000003                          R3=     %3
000004                          R4=     %4
000005                          R5=     %5
000005                          TTY=    %5
000006                          SP=     %6
000007                          PC=     %7
000000                          AC0=    %0
000001                          AC1=    %1
000002                          AC2=    %2
000003                          AC3=    %3
000004                          AC4=    %4
000005                          AC5=    %5
100000                          SW15=   100000
040000                          SW14=    40000
020000                          SW13=    20000
010000                          SW12=    10000
004000                          SW11=     4000
002000                          SW10=     2000
001000                          SW09=     1000
000400                          SW08=      400
170003                          LDUB=   170003
170005                          STA0=   170005
170007                          STQ0=   170007
170006                          MRS=    170006
170004                          LDSC=   170004

                        000000          .=      0                       ;TRAP CATCHER FROM 0 - 776

                        000200          .=      200

000200  000167  000622                  JMP     BEG

                        000760          .=      760
000760  170200                  FLTERR: STFPS   FPS
000762  170367  000034                  STST    FEC
000766  000000                          HALT
000770  000002                          RTI
```

```
                001000                  .=      1000
001000  000000                  ICNT:   0               ;ITERATION COUNT - LH   TEST NO. - PH
001002  000000                  ANS1:   0               ;FIRST ANSWER (SEE CODE)
001004  000000                  ANS2:   0
001006  000000                  ANS3:   0
001010  000000                  ANS4:   0
001012  000000                  ANS5:   0
001014  000000                  ANS6:   0
001016  000000                  ANS7:   0
001020  000000                  ANS8:   0
001022  000000                  FEC:    0               ;FLOATING EXCEPTION CODES
001024  000000                  FEA:    0               ;FLOATING EXECPTION ADDRESS

001026  012706  000600          BEG:    MOV     #600,SP         ;** STACK AT 600 **
001032  012737  001054  000004          MOV     #M1120,@#4      ;FIND OUT WHICH MACHINE THIS IS
001040  005737  177772                  TST     @#177772        ;IS PIRQ THERE?
001044  012767  000006  007352          MOV     #6,YESRT        ;FUDGE IN RTT IF 11/45
001052  000403                          BR      BEGIN

001054  016737  010506  000010  M1120:  MOV     FPTADR,@#10     ;LOAD THE ILLEGAL INSTRUCTION VECTOR
                                                                ; WITH THE ADDRESS OF THE FPU.
                                                                ; THE FPU WILL HANDLE THE BAD OPCODES
001062  012737  000006  000004  BEGIN:  MOV     #6,@#4          ;RESET 4
001070  012706  000600                  MOV     #600,SP
001074  012737  010424  000014          MOV     #YESRT,@#14     ;SET TRACE TRAP VECTOR
001102  012777  011264  010464          MOV     #POWDWN,@DWNVEC
001110  012777  000340  010460          MOV     #340,@DWNVEC+2
001116  012737  011464  000020          MOV     #.IOT,@#20      ;SET UP VECTOR 20
001124  012700  000030                  MOV     #30,R0          ;SET R0 TO VECTOR 30
001130  012720  010566                  MOV     #.TRP,(0)+      ;SET EMT VECTOR
001134  012720  000340                  MOV     #340,(0)+
001140  012720  010426                  MOV     #.EMT,(0)+      ;SET TRAP VECTOR
001144  012710  000340                  MOV     #340,(0)
001150  012777  000760  010412          MOV     #FLTERR,@FPVECT ;LOAD INTERUPT VECTOR
001156  012777  000340  010406          MOV     #340,@FPVECT+2  ;LOCK UP PROCESSOR
001164  005067  177610                  CLR     ICNT
001170  005067  010414                  CLR     LAD
```

```
                        ;*****************************************************************
                        ;TEST 1:        TEST LDEXP (LOAD EXPONENT), FLOATING MODE
                        ;       000000 --> 025252,125252 ==> 040052,125252
                        ;       FPS = 047400,   SRC = M2-R7,    AC = AC2
                        ;*****************************************************************

001174  104400                  SCOPE
001176  000402                  BR      TST1

001200  025252  125252  DAT1:   025252,125252

001204  170127  047417  TST1:   LDFPS   #047417          ;LOAD FLOATING POINT STATUS
001210  172667  177764          LDF     DAT1,   AC2     ;LOAD 025252,125252 INTO AC2
001214  176627  000000  FPI1:   LDEXP   #000000,AC2     ;LOAD 000000 INTO THE EXPONENT OF AC2
001220  170200                  STFPS   FPS             ;STORE FLOATING POINT STATUS
001222  022700  047400          CMP     #047400,FPS     ;CHECK FLOATING POINT STATUS
001226  001401                  BEQ     .+4             ;BRANCH IF OK
001230  104000                  HLT                     ;FPS NOT EQUAL TO 047400

001232  174267  177544          STF     AC2,    ANS1    ;STORE AC2 INTO ANS1,ANS2
001236  022767  040052  177536  CMP     #040052,ANS1    ;CHECK ANS1
001244  001401                  BEQ     .+4             ;BRANCH IF OK
001246  104002                  HLT+2                   ;ANS1 NOT EQUAL TO 040052

001250  022767  125252  177526  CMP     #125252,ANS2    ;CHECK ANS2
001256  001401                  BEQ     .+4             ;BRANCH IF OK
001260  104002                  HLT+2                   ;ANS2 NOT EQUAL TO 125252

                        ;*****************************************************************
                        ;TEST 2:        TEST LDEXP (LOAD EXPONENT), FLOATING MODE
                        ;       000005 --> 137777,177777 ==> 141377,177777
                        ;       FPS = 047410,   SRC = M2-R7,    AC = AC1
                        ;*****************************************************************

001262  104400                  SCOPE
001264  000402                  BR      TST2

001266  137777  177777  DAT2:   137777,177777

001272  170127  047417  TST2:   LDFPS   #047417          ;LOAD FLOATING POINT STATUS
001276  172567  177764          LDF     DAT2,   AC1     ;LOAD 137777,177777 INTO AC1
001302  176527  000005  FPI2:   LDEXP   #000005,AC1     ;LOAD 000005 INTO THE EXPONENT OF AC1
001306  170200                  STFPS   FPS             ;STORE FLOATING POINT STATUS
001310  022700  047410          CMP     #047410,FPS     ;CHECK FLOATING POINT STATUS
001314  001401                  BEQ     .+4             ;BRANCH IF OK
001316  104000                  HLT                     ;FPS NOT EQUAL TO 047410

001320  174167  177456          STF     AC1,    ANS1    ;STORE AC1 INTO ANS1,ANS2
001324  022767  141377  177450  CMP     #141377,ANS1    ;CHECK ANS1
001332  001401                  BEQ     .+4             ;BRANCH IF OK
001334  104002                  HLT+2                   ;ANS1 NOT EQUAL TO 141377

001336  022767  177777  177440  CMP     #177777,ANS2    ;CHECK ANS2
001344  001401                  BEQ     .+4             ;BRANCH IF OK
001346  104002                  HLT+2                   ;ANS2 NOT EQUAL TO 177777
```

```
                        ;*****************************************************************
                        ;TEST 3:        TEST LDEXP (LOAD EXPONENT), FLOATING MODE
                        ;       177601 --> 000000,000000 ==> 000200,000000
                        ;       FPS = 047400,   SRC = M2-R7,    AC = AC2
                        ;*****************************************************************

001350  104400                  SCOPE
001352  000402                  BR      TST3

001354  000000  000000  DAT3:   000000,000000

001360  170127  047417  TST3:   LDFPS   #047417          ;LOAD FLOATING POINT STATUS
001364  172667  177764          LDF     DAT3,   AC2     ;LOAD 000000,000000 INTO AC2
001370  176627  177601  FPI3:   LDEXP   #177601,AC2     ;LOAD 177601 INTO THE EXPONENT OF AC2
001374  170200                  STFPS   FPS             ;STORE FLOATING POINT STATUS
001376  022700  047400          CMP     #047400,FPS     ;CHECK FLOATING POINT STATUS
001402  001401                  BEQ     .+4             ;BRANCH IF OK
001404  104000                  HLT                     ;FPS NOT EQUAL TO 047400

001406  174267  177370          STF     AC2,    ANS1    ;STORE AC2 INTO ANS1,ANS2
001412  022767  000200  177362  CMF     #000200,ANS1    ;CHECK ANS1
001420  001401                  BEQ     .+4             ;BRANCH IF OK
001422  104002                  HLT+2                   ;ANS1 NOT EQUAL TO 000200

001424  022767  000000  177352  CMP     #000000,ANS2    ;CHECK ANS2
001432  001401                  BEQ     .+4             ;BRANCH IF OK
001434  104002                  HLT+2                   ;ANS2 NOT EQUAL TO 000000

                        ;*****************************************************************
                        ;TEST 4:        TEST LDEXP (LOAD EXPONENT), FLOATING MODE
                        ;       000177 --> 000000,000000 ==> 077600,000000
                        ;       FPS = 047400,   SRC = M2-R7,    AC = AC2
                        ;*****************************************************************

001436  104400                  SCOPE
001440  000402                  BR      TST4

001442  000000  000000  DAT4:   000000,000000

001446  170127  047417  TST4:   LDFPS   #047417          ;LOAD FLOATING POINT STATUS
001452  172667  177764          LDF     DAT4,   AC2     ;LOAD 000000,000000 INTO AC2
001456  176627  000177  FPI4:   LDEXP   #000177,AC2     ;LOAD 000177 INTO THE EXPONENT OF AC2
001462  170200                  STFPS   FPS             ;STORE FLOATING POINT STATUS
001464  022700  047400          CMP     #047400,FPS     ;CHECK FLOATING POINT STATUS
001470  001401                  BEQ     .+4             ;BRANCH IF OK
001472  104000                  HLT                     ;FPS NOT EQUAL TO 047400

001474  174267  177302          STF     AC2,    ANS1    ;STORE AC2 INTO ANS1,ANS2
001500  022767  077600  177274  CMP     #077600,ANS1    ;CHECK ANS1
001506  001401                  BEQ     .+4             ;BRANCH IF OK
001510  104002                  HLT+2                   ;ANS1 NOT EQUAL TO 077600

001512  022767  000000  177264  CMP     #000000,ANS2    ;CHECK ANS2
001520  001401                  BEQ     .+4             ;BRANCH IF OK
001522  104002                  HLT+2                   ;ANS2 NOT EQUAL TO 000000
```

```
                        ;*****************************************************************
                        ;TEST 5:        TEST LDEXP (LOAD EXPONENT), FLOATING MODE
                        ;       177777 --> 040000,000020 ==> 037600,000020
                        ;       FPS = 047400,   SRC = M2-R7,    AC = AC0
                        ;*****************************************************************

001524  104400                  SCOPE
001526  000402                  BR      TST5

001530  040000  000020  DAT5:   040000,000020

001534  170127  047417  TST5:   LDFPS   #047417          ;LOAD FLOATING POINT STATUS
001540  172467  177764          LDF     DAT5,   AC0     ;LOAD 040000,000020 INTO AC0
001544  176427  177777  FPI5:   LDEXP   #177777,AC0     ;LOAD 177777 INTO THE EXPONENT OF AC0
001550  170200                  STFPS   FPS             ;STORE FLOATING POINT STATUS
001552  022700  047400          CMP     #047400,FPS     ;CHECK FLOATING POINT STATUS
001556  001401                  BEQ     .+4             ;BRANCH IF OK
001560  104000                  HLT                     ;FPS NOT EQUAL TO 047400

001562  174067  177214          STF     AC0,    ANS1    ;STORE AC0 INTO ANS1,ANS2
001566  022767  037600  177206  CMP     #037600,ANS1    ;CHECK ANS1
001574  001401                  BEQ     .+4             ;BRANCH IF OK
001576  104002                  HLT+2                   ;ANS1 NOT EQUAL TO 037600

001600  022767  000020  177176  CMP     #000020,ANS2    ;CHECK ANS2
001606  001401                  BEQ     .+4             ;BRANCH IF OK
001610  104002                  HLT+2                   ;ANS2 NOT EQUAL TO 000020

                        ;*****************************************************************
                        ;TEST 6:        TEST LDEXP (LOAD EXPONENT), FLOATING MODE
                        ;       177706 --> 125252,125252 ==> 121452,125252
                        ;       FPS = 047410,   SRC = M2-R7,    AC = AC0
                        ;*****************************************************************

001612  104400                  SCOPE
001614  000402                  BR      TST6

001616  125252  125252  DAT6:   125252,125252

001622  170127  047417  TST6:   LDFPS   #047417          ;LOAD FLOATING POINT STATUS
001626  172467  177764          LDF     DAT6,   AC0     ;LOAD 125252,125252 INTO AC0
001632  176427  177706  FPI6:   LDEXP   #177706,AC0     ;LOAD 177706 INTO THE EXPONENT OF AC0
001636  170200                  STFPS   FPS             ;STORE FLOATING POINT STATUS
001640  022700  047410          CMP     #047410,FPS     ;CHECK FLOATING POINT STATUS
001644  001401                  BEQ     .+4             ;BRANCH IF OK
001646  104000                  HLT                     ;FPS NOT EQUAL TO 047410

001650  174067  177126          STF     AC0,    ANS1    ;STORE AC0 INTO ANS1,ANS2
001654  022767  121452  177120  CMP     #121452,ANS1    ;CHECK ANS1
001662  001401                  BEQ     .+4             ;BRANCH IF OK
001664  104002                  HLT+2                   ;ANS1 NOT EQUAL TO 121452

001666  022767  125252  177110  CMP     #125252,ANS2    ;CHECK ANS2
001674  001401                  BEQ     .+4             ;BRANCH IF OK
001676  104002                  HLT+2                   ;ANS2 NOT EQUAL TO 125252
```

```
;*********************************************
;TEST 1:        TEST LDEXP LDAC EXP...      FLOATING MODE
;       000135 --> 052525,052525 ==> 052525,052525
;       FPS = 047400,   SRC = M2-R7,    AC = AC2
;*********************************************

                        .SCOPE
                        .BR     TST1

DAT1:   052525,052525

TST1:   LDFPS   #047417
        LDF     DAT1,AC2        ;LOAD FLOATING POINT STATUS
        LDEXP   #000135,AC2     ;LOAD 052525,052525 INTO AC2
        STFPS   FPS             ;LOAD 000135 INTO THE EXPONENT OF AC2
        STST    FEC             ;STORE FLOATING POINT STATUS
        CMP     FPS,#047400     ;STORE EXCEPTION CODES
        BEQ     .+4             ;CHECK FLOATING POINT STATUS
        HLT                     ;BRANCH IF OK
                                ;FPS NOT EQUAL TO 047400

        STF     AC2,ANS1        ;STORE AC2 INTO ANS1,ANS2
        CMP     #067325,ANS1    ;CHECK ANS1
        BEQ     .+4             ;BRANCH IF OK
        HLT     ^O2             ;ANS1 NOT EQUAL TO 067325

        CMP     #052525,ANS2    ;CHECK ANS2
        BEQ     .+4             ;BRANCH IF OK
        HLT     ^O2             ;ANS2 NOT EQUAL TO 052525

;*********************************************
;TEST 10:       TEST LDEXP LDAC EXPONENT), FLOATING MODE
;       000200 --> 140425,125252 ==> 100025,125252
;       FPS = 147416,   SRC = M2-R7,    AC = AC0
;       FEC = 10,       FEA = FPI1C
;*********************************************

                        .SCOPE

DAT1C:

TST1C:

FPI1C:  LDF     DAT1C,AC0       ;LOAD FLOATING POINT STATUS
        LDEXP   #000200,AC0     ;LOAD 140425,125252 INTO AC0
        STFPS   FPS             ;LOAD 000200 INTO THE EXPONENT OF AC0
        STST    FEC             ;STORE FLOATING POINT STATUS
        CMP     #147416,FPS     ;STORE EXCEPTION CODES
        BEQ     .+4             ;CHECK FLOATING POINT STATUS
        HLT                     ;BRANCH IF OK
                                ;FPS NOT EQUAL TO 147416

        CMP     #10.,   FEC     ;CHECK FLOATING EXCEPTION CODE
        BEQ     .+4             ;BRANCH IF OK
        HLT                     ;FEC NOT EQUAL TO 10

        CMP     #FPI1C, FEA     ;CHECK FLOATING EXCEPTION ADDRESS
        BEQ     .+4             ;BRANCH IF OK
        HLT                     ;FEA NOT EQUAL TO FPI1C
```

```
002054  174067  176722          STF     AC0,    ANS1    ;STORE AC0 INTO ANS1,ANS2
002060  022767  100025  176714  CMP     #100025,ANS1    ;CHECK ANS1
002066  001401                  BEQ     .+4             ;BRANCH IF OK
002070  104002                  HLT+2                   ;ANS1 NOT EQUAL TO 100025

002072  022767  125252  176704  CMP     #125252,ANS2    ;CHECK ANS2
002100  001401                  BEQ     .+4             ;BRANCH IF OK
002102  104002                  HLT+2                   ;ANS2 NOT EQUAL TO 125252

                        ;*****************************************************************
                        ;TEST 11:       TEST LDEXP (LOAD EXPONENT), FLOATING MODE
                        ;       177601 --> 050505,050505 ==> 000305,050505
                        ;       FPS = 047400,   SRC = M2-R7,    AC = AC2
                        ;*****************************************************************

002104  104400                  SCOPE
002106  000402                  BR      TST11

002110  050505  050505  DAT11:  050505,050505

002114  170127  047417  TST11:  LDFPS   #047417          ;LOAD FLOATING POINT STATUS
002120  172667  177764          LDF     DAT11,  AC2     ;LOAD 050505,050505 INTO AC2
002124  176627  177601  FPI11:  LDEXP   #177601,AC2     ;LOAD 177601 INTO THE EXPONENT OF AC2
002130  170200                  STFPS   FPS             ;STORE FLOATING POINT STATUS
002132  022700  047400          CMP     #047400,FPS     ;CHECK FLOATING POINT STATUS
002136  001401                  BEQ     .+4             ;BRANCH IF OK
002140  104000                  HLT                     ;FPS NOT EQUAL TO 047400

002142  174267  176634          STF     AC2,    ANS1    ;STORE AC2 INTO ANS1,ANS2
002146  022767  000305  176626  CMP     #000305,ANS1    ;CHECK ANS1
002154  001401                  BEQ     .+4             ;BRANCH IF OK
002156  104002                  HLT+2                   ;ANS1 NOT EQUAL TO 000305

002160  022767  050505  176616  CMP     #050505,ANS2    ;CHECK ANS2
002166  001401                  BEQ     .+4             ;BRANCH IF OK
002170  104002                  HLT+2                   ;ANS2 NOT EQUAL TO 050505

                        ;*****************************************************************
                        ;TEST 12:       TEST LDEXP (LOAD EXPONENT), FLOATING MODE
                        ;       177600 --> 040525,125252 ==> 000125,125252
                        ;       FPS = 147404,   SRC = M2-R7,    AC = AC0
                        ;       FEC = 12,       FEA = FPI12
                        ;*****************************************************************

002172  104400                  SCOPE
002174  000402                  BR      TST12

002176  040525  125252  DAT12:  040525,125252

002202  170127  047417  TST12:  LDFPS   #047417          ;LOAD FLOATING POINT STATUS
002206  172467  177764          LDF     DAT12,  AC0     ;LOAD 040525,125252 INTO AC0
002212  176427  177600  FPI12:  LDEXP   #177600,AC0     ;LOAD 177600 INTO THE EXPONENT OF AC0
002216  170200                  STFPS   FPS             ;STORE FLOATING POINT STATUS
002220  170367  176576          STST    FEC             ;STORE EXCEPTION CODES
002224  022700  147404          CMP     #147404,FPS     ;CHECK FLOATING POINT STATUS
```

```
002230  001401                          BEQ     .+4             ;BRANCH IF OK
002232  104000                          HLT                     ;FPS NOT EQUAL TO 147404

002234  022767  000012  176560          CMP     #12.,   FEC     ;CHECK FLOATING EXCEPTION CODE
002242  001401                          BEQ     .+4             ;BRANCH IF OK
002244  104000                          HLT                     ;FEC NOT EQUAL TO 12

002246  022767  002212  176550          CMP     #FPI12, FEA     ;CHECK FLOATING EXCEPTION ADDRESS
002254  001401                          BEQ     .+4             ;BRANCH IF OK
002256  104000                          HLT                     ;FEA NOT EQUAL TO FPI12

002260  174067  176516                  STF     AC0,    ANS1    ;STORE AC0 INTO ANS1,ANS2
002264  022767  000125  176510          CMP     #000125,ANS1    ;CHECK ANS1
002272  001401                          BEQ     .+4             ;BRANCH IF OK
002274  104002                          HLT+2                   ;ANS1 NOT EQUAL TO 000125

002276  022767  125252  176500          CMP     #125252,ANS2    ;CHECK ANS2
002304  001401                          BEQ     .+4             ;BRANCH IF OK
002306  104002                          HLT+2                   ;ANS2 NOT EQUAL TO 125252

                        ;*****************************************************************
                        ;TEST 13:       TEST LDEXP (LOAD EXPONENT), FLOATING MODE
                        ;       177575 --> 050505,050505 ==> 077305,050505
                        ;       FPS = 147400,   SRC = M2-R7,    AC = AC2
                        ;       FEC = 12,       FEA = FPI13
                        ;*****************************************************************

002310  104400                          SCOPE
002312  000402                          BR      TST13

002314  050505  050505          DAT13:  050505,050505

002320  170127  047417          TST13:  LDFPS   #047417         ;LOAD FLOATING POINT STATUS
002324  172667  177764                  LDF     DAT13,  AC2     ;LOAD 050505,050505 INTO AC2
002330  176627  177575          FPI13:  LDEXP   #177575,AC2     ;LOAD 177575 INTO THE EXPONENT OF AC2
002334  170200                          STFPS   FPS             ;STORE FLOATING POINT STATUS
002336  170367  176460                  STST    FEC             ;STORE EXCEPTION CODES
002342  022700  147400                  CMP     #147400,FPS     ;CHECK FLOATING POINT STATUS
002346  001401                          BEQ     .+4             ;BRANCH IF OK
002350  104000                          HLT                     ;FPS NOT EQUAL TO 147400

002352  022767  000012  176442          CMP     #12.,   FEC     ;CHECK FLOATING EXCEPTION CODE
002360  001401                          BEQ     .+4             ;BRANCH IF OK
002362  104000                          HLT                     ;FEC NOT EQUAL TO 12

002364  022767  002330  176432          CMP     #FPI13, FEA     ;CHECK FLOATING EXCEPTION ADDRESS
002372  001401                          BEQ     .+4             ;BRANCH IF OK
002374  104000                          HLT                     ;FEA NOT EQUAL TO FPI13

002376  174267  176400                  STF     AC2,    ANS1    ;STORE AC2 INTO ANS1,ANS2
002402  022767  077305  176372          CMP     #077305,ANS1    ;CHECK ANS1
002410  001401                          BEQ     .+4             ;BRANCH IF OK
002412  104002                          HLT+2                   ;ANS1 NOT EQUAL TO 077305

002414  022767  050505  176362          CMP     #050505,ANS2    ;CHECK ANS2
002422  001401                          BEQ     .+4             ;BRANCH IF OK
```

```
002434  104002                  HLT+2                   ;ANS2 NOT EQUAL TO 050505

                        ;*****************************************************************
                        ;TEST 14:       TEST LDEXP (LOAD EXPONENT), FLOATING MODE
                        ;       000201 --> 140425,125252 ==> 100225,125252
                        ;       FPS = 046412,   SRC = M2-R7,    AC = AC0
                        ;*****************************************************************

002436  104400                  SCOPE
002440  000402                  BR      TST14

002442  140425  125252  DAT14:  140425,125252

002446  170127  046405  TST14:  LDFPS   #046405          ;LOAD FLOATING POINT STATUS
002452  172467  177764          LDF     DAT14,  AC0     ;LOAD 140425,125252 INTO AC0
002456  176427  000201  FPI14:  LDEXP   #000201,AC0     ;LOAD 000201 INTO THE EXPONENT OF AC0
002462  170200                  STFPS   FPS             ;STORE FLOATING POINT STATUS
002464  022700  046412          CMP     #046412,FPS     ;CHECK FLOATING POINT STATUS
002470  001401                  BEQ     .+4             ;BRANCH IF OK
002472  104000                  HLT                     ;FPS NOT EQUAL TO 046412

002474  174067  176312          STF     AC0,    ANS1    ;STORE AC0 INTO ANS1,ANS2
002500  022767  100225  176304  CMP     #100225,ANS1    ;CHECK ANS1
002506  001401                  BEQ     .+4             ;BRANCH IF OK
002510  104002                  HLT+2                   ;ANS1 NOT EQUAL TO 100225

002512  022767  125252  176274  CMP     #125252,ANS2    ;CHECK ANS2
002520  001401                  BEQ     .+4             ;BRANCH IF OK
002522  104002                  HLT+2                   ;ANS2 NOT EQUAL TO 125252

                        ;*****************************************************************
                        ;TEST 15:       TEST LDEXP (LOAD EXPONENT), FLOATING MODE
                        ;       177576 --> 040525,125252 ==> 000000,000000
                        ;       FPS = 045404,   SRC = M2-R7,    AC = AC2
                        ;*****************************************************************

002524  104400                  SCOPE
002526  000402                  BR      TST15

002530  040525  125252  DAT15:  040525,125252

002534  170127  045417  TST15:  LDFPS   #045417          ;LOAD FLOATING POINT STATUS
002540  172667  177764          LDF     DAT15,  AC2     ;LOAD 040525,125252 INTO AC2
002544  176627  177576  FPI15:  LDEXP   #177576,AC2     ;LOAD 177576 INTO THE EXPONENT OF AC2
002550  170200                  STFPS   FPS             ;STORE FLOATING POINT STATUS
002552  022700  045404          CMP     #045404,FPS     ;CHECK FLOATING POINT STATUS
002556  001401                  BEQ     .+4             ;BRANCH IF OK
002560  104000                  HLT                     ;FPS NOT EQUAL TO 045404

002562  174267  176224          STF     AC2,    ANS1    ;STORE AC2 INTO ANS1,ANS2
002566  022767  000000  176216  CMP     #000000,ANS1    ;CHECK ANS1
002574  001401                  BEQ     .+4             ;BRANCH IF OK
002576  104002                  HLT+2                   ;ANS1 NOT EQUAL TO 000000

002600  022767  000000  176206  CMP     #000000,ANS2    ;CHECK ANS2
002606  001401                  BEQ     .+4             ;BRANCH IF OK
```

```
002630  104002                          HLT+2                   ;ANS2 NOT EQUAL TO 000000

                        ;*****************************************************************
                        ;TEST 16:       TEST LDEXP (LOAD EXPONENT), FLOATING MODE
                        ;       177706 --> 125252,125252 ==> 121452,125252
                        ;       FPS = 047410,   SRC = MD-R2,    AC = AC0
                        ;*****************************************************************

002632  104400                          SCOPE
002634  000402                          BR      TST16

002636  125252  125252        DAT16:  125252,125252

002642  170127  047417        TST16:  LDFPS   #047417         ;LOAD FLOATING POINT STATUS
002646  172467  177764                LDF     DAT16,  AC0     ;LOAD 125252,125252 INTO AC0
002654  012702  177706                MOV     #177706,R2      ;PUT EXPONENT IN R2
002656  176402                FPI16:  LDEXP   R2,     AC0     ;LOAD 177706 INTO THE EXPONENT OF AC0
002660  170200                        STFPS   FPS             ;STORE FLOATING POINT STATUS
002662  022700  047410                CMP     #047410,FPS     ;CHECK FLOATING POINT STATUS
002666  001401                        BEQ     .+4             ;BRANCH IF OK
002670  104000                        HLT                     ;FPS NOT EQUAL TO 047410

002672  174067  176134                STF     AC0,    ANS1    ;STORE AC0 INTO ANS1,ANS2
002676  022767  121452  176126        CMP     #121452,ANS1    ;CHECK ANS1
002704  001401                        BEQ     .+4             ;BRANCH IF OK
002706  104002                        HLT+2                   ;ANS1 NOT EQUAL TO 121452

002710  022767  125252  176116        CMP     #125252,ANS2    ;CHECK ANS2
002716  001401                        BEQ     .+4             ;BRANCH IF OK
002720  104002                        HLT+2                   ;ANS2 NOT EQUAL TO 125252

                        ;*****************************************************************
                        ;TEST 17:       TEST LDEXP (LOAD EXPONENT), FLOATING MODE
                        ;       000135 --> 052525,052525 ==> 067325,052525
                        ;       FPS = 047400,   SRC = MD-R3,    AC = AC2
                        ;*****************************************************************

002722  104400                          SCOPE
002724  000402                          BR      TST17

002726  052525  052525        DAT17:  052525,052525

002732  170127  047417        TST17:  LDFPS   #047417         ;LOAD FLOATING POINT STATUS
002736  172667  177764                LDF     DAT17,  AC2     ;LOAD 052525,052525 INTO AC2
002744  012703  000135                MOV     #000135,R3      ;PUT EXPONENT IN R3
002750  176603                FPI17:  LDEXP   R3,     AC2     ;LOAD 000135 INTO THE EXPONENT OF AC2
002752  170200                        STFPS   FPS             ;STORE FLOATING POINT STATUS
002754  022700  047400                CMP     #047400,FPS     ;CHECK FLOATING POINT STATUS
002760  001401                        BEQ     .+4             ;BRANCH IF OK
002762  104000                        HLT                     ;FPS NOT EQUAL TO 047400

002764  174267  176044                STF     AC2,    ANS1    ;STORE AC2 INTO ANS1,ANS2
002770  022767  067325  176036        CMP     #067325,ANS1    ;CHECK ANS1
002776  001401                        BEQ     .+4             ;BRANCH IF OK
003000  104002                        HLT+2                   ;ANS1 NOT EQUAL TO 067325
```

```
002750  022767  052525  176026          CMP     #052525,ANS2    ;CHECK ANS2
002756  001401                          BEQ     .+4             ;BRANCH IF OK
002760  104002                          HLT+2                   ;ANS2 NOT EQUAL TO 052525

                        ;*****************************************************************
                        ;TEST 20:       TEST STEXP (STORE EXPONENT), FLOATING MODE
                        ;       000000,000000 --> 177600
                        ;       FPS = 047410,   FCC = 10,       AC = AC0,       DST = M6-R7
                        ;*****************************************************************

002762  104400                          SCOPE
002764  000402                          BR      TST20

002766  000000  000000        DAT20:  000000,000000

002772  170127  047417        TST20:  LDFPS   #047417         ;LOAD FLOATING POINT STATUS
002776  172467  177764                LDF     DAT20,  AC0     ;LOAD 000000,000000 INTO AC0
003002  175067  175774        FPI20:  STEXP   AC0,    ANS1    ;STORE THE EXPONENT OF AC0 IN ANS1
003006  013767  177776  175770        MOV     @#PS,   ANS2    ;GET CPU STATUS
003014  042767  177760  175762        BIC     #177760,ANS2    ;SAVE CONDITION CODES
003022  170200                        STFPS   FPS             ;STORE FLOATING POINT STATUS
003024  022700  047410                CMP     #047410,FPS     ;CHECK FLOATING POINT STATUS
003030  001401                        BEQ     .+4             ;BRANCH IF OK
003032  104000                        HLT                     ;FPS NOT EQUAL TO 047410

003034  022767  177600  175740        CMP     #177600,ANS1    ;CHECK ANS1
003042  001401                        BEQ     .+4             ;BRANCH IF OK
003044  104001                        HLT+1                   ;ANS1 NOT EQUAL TO 177600

003046  022767  000010  175730        CMP     #10.    ANS2    ;CHECK ANS2
003054  001401                        BEQ     .+4             ;BRANCH IF OK
003056  104002                        HLT+2                   ;WRONG CONDITION CODES!

                        ;*****************************************************************
                        ;TEST 21:       TEST STEXP (STORE EXPONENT), FLOATING MODE
                        ;       140134,034343 --> 000000
                        ;       FPS = 047404,   FCC = 4,        AC = AC2,       DST = M6-R7
                        ;*****************************************************************

003060  104400                          SCOPE
003062  000402                          BR      TST21

003064  140134  034343        DAT21:  140134,034343

003070  170127  047417        TST21:  LDFPS   #047417         ;LOAD FLOATING POINT STATUS
003074  172667  177764                LDF     DAT21,  AC2     ;LOAD 140134,034343 INTO AC2
003100  175267  175676        FPI21:  STEXP   AC2,    ANS1    ;STORE THE EXPONENT OF AC2 IN ANS1
003104  013767  177776  175672        MOV     @#PS,   ANS2    ;GET CPU STATUS
003112  042767  177760  175664        BIC     #177760,ANS2    ;SAVE CONDITION CODES
003120  170200                        STFPS   FPS             ;STORE FLOATING POINT STATUS
003122  022700  047404                CMP     #047404,FPS     ;CHECK FLOATING POINT STATUS
003126  001401                        BEQ     .+4             ;BRANCH IF OK
003130  104000                        HLT                     ;FPS NOT EQUAL TO 047404

003132  022767  000000  175642        CMP     #000000,ANS1    ;CHECK ANS1
003140  001401                        BEQ     .+4             ;BRANCH IF OK
```

```
003142  104001                          HLT+1                   ;ANS1 NOT EQUAL TO 000000

003144  022767  000004  175632          CMP     #4,     ANS2    ;CHECK ANS2
003152  001401                          BEQ     .+4             ;BRANCH IF OK
003154  104002                          HLT+2                   ;WRONG CONDITION CODES!

                        ;*****************************************************************
                        ;TEST 22:       TEST STEXP (STORE EXPONENT), FLOATING MODE
                        ;       040200,125252 --> 000001
                        ;       FPS = 047400,   FCC = 0,        AC = AC0,       DST = M6-R7
                        ;*****************************************************************

003156  104400                          SCOPE
003160  000402                          BR      TST22

003162  040200  125252          DAT22:  040200,125252

003166  170127  047417          TST22:  LDFPS   #047417         ;LOAD FLOATING POINT STATUS
003172  172467  177764                  LDF     DAT22,  AC0     ;LOAD 040200,125252 INTO AC0
003176  175067  175600          FPI22:  STEXP   AC0,    ANS1    ;STORE THE EXPONENT OF AC0 IN ANS1
003202  013767  177776  175574          MOV     @#PS,   ANS2    ;GET CPU STATUS
003210  042767  177760  175566          BIC     #177760,ANS2    ;SAVE CONDITION CODES
003216  170200                          STFPS   FPS             ;STORE FLOATING POINT STATUS
003220  022700  047400                  CMP     #047400,FPS     ;CHECK FLOATING POINT STATUS
003224  001401                          BEQ     .+4             ;BRANCH IF OK
003226  104000                          HLT                     ;FPS NOT EQUAL TO 047400

003230  022767  000001  175544          CMP     #000001,ANS1    ;CHECK ANS1
003236  001401                          BEQ     .+4             ;BRANCH IF OK
003240  104001                          HLT+1                   ;ANS1 NOT EQUAL TO 000001

003242  022767  000000  175534          CMP     #0,     ANS2    ;CHECK ANS2
003250  001401                          BEQ     .+4             ;BRANCH IF OK
003252  104002                          HLT+2                   ;WRONG CONDITION CODES!

                        ;*****************************************************************
                        ;TEST 23:       TEST STEXP (STORE EXPONENT), FLOATING MODE
                        ;       042525,052525 --> 000012
                        ;       FPS = 047400,   FCC = 0,        AC = AC0,       DST = M6-R7
                        ;*****************************************************************

003254  104400                          SCOPE
003256  000402                          BR      TST23

003260  042525  052525          DAT23:  042525,052525

003264  170127  047417          TST23:  LDFPS   #047417         ;LOAD FLOATING POINT STATUS
003270  172467  177764                  LDF     DAT23,  AC0     ;LOAD 042525,052525 INTO AC0
003274  175067  175502          FPI23:  STEXP   AC0,    ANS1    ;STORE THE EXPONENT OF AC0 IN ANS1
003300  013767  177776  175476          MOV     @#PS,   ANS2    ;GET CPU STATUS
003306  042767  177760  175470          BIC     #177760,ANS2    ;SAVE CONDITION CODES
003314  170200                          STFPS   FPS             ;STORE FLOATING POINT STATUS
003316  022700  047400                  CMP     #047400,FPS     ;CHECK FLOATING POINT STATUS
003322  001401                          BEQ     .+4             ;BRANCH IF OK
003324  104000                          HLT                     ;FPS NOT EQUAL TO 047400
```

```
003326  022767  000012  175446          CMP     #000012,ANS1    ;CHECK ANS1
003334  001401                          BEQ     .+4             ;BRANCH IF OK
003336  104001                          HLT+1                   ;ANS1 NOT EQUAL TO 000012

003340  022767  000000  175436          CMP     #0,     ANS2    ;CHECK ANS2
003346  001401                          BEQ     .+4             ;BRANCH IF OK
003350  104002                          HLT+2                   ;WRONG CONDITION CODES!

                        ;*****************************************************************
                        ;TEST 24:       TEST STEXP (STORE EXPONENT), FLOATING MODE
                        ;       125252,125252 --> 177725
                        ;       FPS = 047410,   FCC = 10,       AC = AC2,       DST = M6-R7
                        ;*****************************************************************

003352  104400                          SCOPE
003354  000402                          BR      TST24

003356  125252  125252          DAT24:  125252,125252

003362  170127  047417          TST24:  LDFPS   #047417         ;LOAD FLOATING POINT STATUS
003366  172667  177764                  LDF     DAT24,  AC2     ;LOAD 125252,125252 INTO AC2
003372  175267  175404          FPI24:  STEXP   AC2,    ANS1    ;STORE THE EXPONENT OF AC2 IN ANS1
003376  013767  177776  175400          MOV     @#PS,   ANS2    ;GET CPU STATUS
003404  042767  177760  175372          BIC     #177760,ANS2    ;SAVE CONDITION CODES
003412  170200                          STFPS   FPS             ;STORE FLOATING POINT STATUS
003414  022700  047410                  CMP     #047410,FPS     ;CHECK FLOATING POINT STATUS
003420  001401                          BEQ     .+4             ;BRANCH IF OK
003422  104000                          HLT                     ;FPS NOT EQUAL TO 047410

003424  022767  177725  175350          CMP     #177725,ANS1    ;CHECK ANS1
003432  001401                          BEQ     .+4             ;BRANCH IF OK
003434  104001                          HLT+1                   ;ANS1 NOT EQUAL TO 177725

003436  022767  000010  175340          CMP     #10,    ANS2    ;CHECK ANS2
003444  001401                          BEQ     .+4             ;BRANCH IF OK
003446  104002                          HLT+2                   ;WRONG CONDITION CODES!

                        ;*****************************************************************
                        ;TEST 25:       TEST STEXP (STORE EXPONENT), FLOATING MODE
                        ;       052525,052525 --> 000052
                        ;       FPS = 047400,   FCC = 00,       AC = AC0,       DST = M6-R7
                        ;*****************************************************************

003450  104400                          SCOPE
003452  000402                          BR      TST25

003454  052525  052525          DAT25:  052525,052525

003460  170127  047417          TST25:  LDFPS   #047417         ;LOAD FLOATING POINT STATUS
003464  172467  177764                  LDF     DAT25,  AC0     ;LOAD 052525,052525 INTO AC0
003470  175067  175306          FPI25:  STEXP   AC0,    ANS1    ;STORE THE EXPONENT OF AC0 IN ANS1
003474  013767  177776  175302          MOV     @#PS,   ANS2    ;GET CPU STATUS
003502  042767  177760  175274          BIC     #177760,ANS2    ;SAVE CONDITION CODES
003510  170200                          STFPS   FPS             ;STORE FLOATING POINT STATUS
003512  022700  047400                  CMP     #047400,FPS     ;CHECK FLOATING POINT STATUS
003516  001401                          BEQ     .+4             ;BRANCH IF OK
```

```
003520  104000                          HLT                     ;FPS NOT EQUAL TO 047400

003522  022767  000052  175252          CMP     #000052,ANS1    ;CHECK ANS1
003530  001401                          BEQ     .+4             ;BRANCH IF OK
003532  104001                          HLT+1                   ;ANS1 NOT EQUAL TO 000052

003534  022767  000000  175242          CMP     #00,    ANS2    ;CHECK ANS2
003542  001401                          BEQ     .+4             ;BRANCH IF OK
003544  104002                          HLT+2                   ;WRONG CONDITION CODES!

                        ;*****************************************************************
                        ;TEST 26:       TEST STEXP (STORE EXPONENT), FLOATING MODE
                        ;       177777,177777 --> 000177
                        ;       FPS = 047400,   FCC = 00,       AC = AC3,       DST = M6-R7
                        ;*****************************************************************

003546  104400                          SCOPE
003550  000402                          BR      TST26

003552  177777  177777          DAT26:  177777,177777

003556  170127  047417          TST26:  LDFPS   #047417         ;LOAD FLOATING POINT STATUS
003562  172767  177764                  LDF     DAT26,  AC3     ;LOAD 177777,177777 INTO AC3
003566  175367  175210          FPI26:  STEXP   AC3,    ANS1    ;STORE THE EXPONENT OF AC3 IN ANS1
003572  013767  177776  175204          MOV     @#PS,   ANS2    ;GET CPU STATUS
003600  042767  177760  175176          BIC     #177760,ANS2    ;SAVE CONDITION CODES
003606  170200                          STFPS   FPS             ;STORE FLOATING POINT STATUS
003610  022700  047400                  CMP     #047400,FPS     ;CHECK FLOATING POINT STATUS
003614  001401                          BEQ     .+4             ;BRANCH IF OK
003616  104000                          HLT                     ;FPS NOT EQUAL TO 047400

003620  022767  000177  175154          CMP     #000177,ANS1    ;CHECK ANS1
003626  001401                          BEQ     .+4             ;BRANCH IF OK
003630  104001                          HLT+1                   ;ANS1 NOT EQUAL TO 000177

003632  022767  000000  175144          CMP     #00,    ANS2    ;CHECK ANS2
003640  001401                          BEQ     .+4             ;BRANCH IF OK
003642  104002                          HLT+2                   ;WRONG CONDITION CODES!

                        ;*****************************************************************
                        ;TEST 27:       TEST STEXP (STORE EXPONENT), FLOATING MODE
                        ;       000200,000000 --> 177601
                        ;       FPS = 047410,   FCC = 10,       AC = AC1,       DST = M6-R7
                        ;*****************************************************************

003644  104400                          SCOPE
003646  000402                          BR      TST27

003650  000200  000000          DAT27:  000200,000000

003654  170127  047417          TST27:  LDFPS   #047417         ;LOAD FLOATING POINT STATUS
003660  172567  177764                  LDF     DAT27,  AC1     ;LOAD 000200,000000 INTO AC1
003664  175167  175112          FPI27:  STEXP   AC1,    ANS1    ;STORE THE EXPONENT OF AC1 IN ANS1
003670  013767  177776  175106          MOV     @#PS,   ANS2    ;GET CPU STATUS
003676  042767  177760  175100          BIC     #177760,ANS2    ;SAVE CONDITION CODES
003704  170200                          STFPS   FPS             ;STORE FLOATING POINT STATUS
```

```
003706  022700  047410              CMP     #047410,FPS     ;CHECK FLOATING POINT STATUS
003712  001401                      BEQ     .+4             ;BRANCH IF OK
003714  104000                      HLT                     ;FPS NOT EQUAL TO 047410

003716  022767  177601  175056      CMP     #177601,ANS1    ;CHECK ANS1
003724  001401                      BEQ     .+4             ;BRANCH IF OK
003726  104001                      HLT+1                   ;ANS1 NOT EQUAL TO 177601

003730  022767  000010  175046      CMP     #10.    ANS2    ;CHECK ANS2
003736  001401                      BEQ     .+4             ;BRANCH IF OK
003740  104002                      HLT+2                   ;WRONG CONDITION CODES!

                        ;*****************************************************************
                        ;TEST 30:       TEST STEXP (STORE EXPONENT), FLOATING MODE
                        ;       125252,125252 --> 177725
                        ;       FPS = 047410,   FCC = 10,       AC = AC3,       DST = MD-R1
                        ;*****************************************************************

003742  104400                      SCOPE
003744  000402                      BR      TST30

003746  125252  125252      DAT30:  125252,125252

003752  170127  047417      TST30:  LDFPS   #047417         ;LOAD FLOATING POINT STATUS
003756  172767  177764              LDF     DAT30,  AC3     ;LOAD 125252,125252 INTO AC3
003762  175301                      STEXP   AC3,    R1      ;STORE THE EXPONENT OF AC3 IN R1
003764  013767  177776  175012      MOV     @#PS,   ANS2    ;GET CPU STATUS
003772  042767  177760  175004      BIC     #177760,ANS2    ;SAVE CONDITION CODES
004000  010167  174776              MOV     R1,     ANS1    ;STORE EXPONENT FOR TYPING
004004  170200                      STFPS   FPS             ;STORE FLOATING POINT STATUS
004006  022700  047410              CMP     #047410,FPS     ;CHECK FLOATING POINT STATUS
004012  001401                      BEQ     .+4             ;BRANCH IF OK
004014  104000                      HLT                     ;FPS NOT EQUAL TO 047410

004016  022767  177725  174756      CMP     #177725,ANS1    ;CHECK ANS1
004024  001401                      BEQ     .+4             ;BRANCH IF OK
004026  104001                      HLT+1                   ;ANS1 NOT EQUAL TO 177725

004030  022767  000010  174746      CMP     #10,    ANS2    ;CHECK ANS2
004036  001401                      BEQ     .+4             ;BRANCH IF OK
004040  104002                      HLT+2                   ;WRONG CONDITION CODES!

                        ;*****************************************************************
                        ;TEST 31:       TEST STEXP (STORE EXPONENT), FLOATING MODE
                        ;       052525,052525 --> 000052
                        ;       FPS = 047400,   FCC = 00,       AC = AC2,       DST = MD-R4
                        ;*****************************************************************

004042  104400                      SCOPE
004044  000402                      BR      TST31

004046  052525  052525      DAT31:  052525,052525

004052  170127  047417      TST31:  LDFPS   #047417         ;LOAD FLOATING POINT STATUS
004056  172667  177764              LDF     DAT31,  AC2     ;LOAD 052525,052525 INTO AC2
004062  175204                      STEXP   AC2,    R4      ;STORE THE EXPONENT OF AC2 IN R4
```

```
004064  013767  177776  174712          MOV     @#PS,   ANS2    ;GET CPU STATUS
004072  042767  177760  174704          BIC     #177760,ANS2    ;SAVE CONDITION CODES
004100  010467  174676                  MOV     R4,     ANS1    ;STORE EXPONENT FOR TYPING
004104  170200                          STFPS   FPS             ;STORE FLOATING POINT STATUS
004106  022700  047400                  CMP     #047400,FPS     ;CHECK FLOATING POINT STATUS
004112  001401                          BEQ     .+4             ;BRANCH IF OK
004114  104000                          HLT                     ;FPS NOT EQUAL TO 047400

004116  022767  000052  174656          CMP     #000052,ANS1    ;CHECK ANS1
004124  001401                          BEQ     .+4             ;BRANCH IF OK
004126  104001                          HLT+1                   ;ANS1 NOT EQUAL TO 000052

004130  022767  000000  174646          CMP     #00,    ANS2    ;CHECK ANS2
004136  001401                          BEQ     .+4             ;BRANCH IF OK
004140  104002                          HLT+2                   ;WRONG CONDITION CODES!

                        ;*****************************************************************
                        ;TEST 32:       TEST STEXP (STORE EXPONENT), FLOATING MODE
                        ;       143125,052525 --> 000014
                        ;       FPS = 047400,   FCC = 00,       AC = AC0,       DST = M2-R7
                        ;*****************************************************************

004142  104400                          SCOPE
004144  000402                          BR      TST32

004146  143125  052525          DAT32:  143125,052525

004152  170127  047417          TST32:  LDFPS   #047417         ;LOAD FLOATING POINT STATUS
004156  172467  177764                  LDF     DAT32,  AC0     ;LOAD 143125,052525 INTO AC0
004162  005067  000002                  CLR     ANR32
004166  175027                  FPI32:  STEXP   AC0,    (PC)+   ;STORE THE EXPONENT OF AC0 IN .+2
004170  000000                  ANR32:  0                       ;THE EXPONENT GOES HERE
004172  000403                          BR      .+10
004174  000000                          HALT
004176  000000                          HALT
004200  000000                          HALT
004202  013767  177776  174574          MOV     @#PS,   ANS2    ;GET CPU STATUS
004210  042767  177760  174566          BIC     #177760,ANS2    ;SAVE CONDITION CODES
004216  170200                          STFPS   FPS             ;STORE FLOATING POINT STATUS
004220  022700  047400                  CMP     #047400,FPS     ;CHECK FLOATING POINT STATUS
004224  001401                          BEQ     .+4             ;BRANCH IF OK
004226  104000                          HLT                     ;FPS NOT EQUAL TO 047400

004230  016767  177734  174544          MOV     ANR32,ANS1
004236  022767  000014  174536          CMP     #000014,ANS1    ;CHECK ANS1
004244  001401                          BEQ     .+4             ;BRANCH IF OK
004246  104001                          HLT+1                   ;ANS1 NOT EQUAL TO 000014

004250  022767  000000  174526          CMP     #00,    ANS2    ;CHECK ANS2
004256  001401                          BEQ     .+4             ;BRANCH IF OK
004260  104002                          HLT+2                   ;WRONG CONDITION CODES!

                        ;*****************************************************************
                        ;TEST 33:       TEST STEXP (STORE EXPONENT), FLOATING MODE
                        ;       032615,125252 --> 177753
                        ;       FPS = 047410,   FCC = 10,       AC = AC1,       DST = M2-R7
```

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

004262  104400                    SCOPE
004264  000402                    BR      TST33

004266  032615  125252    DAT33:  032615,125252

004272  170127  047417    TST33:  LDFPS   #047417         ;LOAD FLOATING POINT STATUS
004276  172567  177764            LDF     DAT33,  AC1     ;LOAD 032615,125252 INTO AC1
004302  005067  000002            CLR     ANR33
004306  175127              FPI33:  STEXP   AC1,    (PC)+   ;STORE THE EXPONENT OF AC1 IN .+2
004310  000000              ANR33:  0                       ;THE EXPONENT GOES HERE
004312  000403                      BR      .+10
004314  000000                      HALT
004316  000000                      HALT
004320  000000                      HALT
004322  013767  177776  174454    MOV     @#PS,   ANS2    ;GET CPU STATUS
004330  042767  177760  174446    BIC     #177760,ANS2    ;SAVE CONDITION CODES
004336  170200                    STFPS   FPS             ;STORE FLOATING POINT STATUS
004340  022700  047410            CMP     #047410,FPS     ;CHECK FLOATING POINT STATUS
004344  001401                    BEQ     .+4             ;BRANCH IF OK
004346  104000                    HLT                     ;FPS NOT EQUAL TO 047410

004350  016767  177734  174424    MOV     ANR33,ANS1
004356  022767  177753  174416    CMP     #177753,ANS1    ;CHECK ANS1
004364  001401                    BEQ     .+4             ;BRANCH IF OK
004366  104001                    HLT+1                   ;ANS1 NOT EQUAL TO 177753

004370  022767  000010  174406    CMP     #10,    ANS2    ;CHECK ANS2
004376  001401                    BEQ     .+4             ;BRANCH IF OK
004400  104002                    HLT+2                   ;WRONG CONDITION CODES!

                        ;*****************************************************************************
                        ;TEST 34:       TEST LDEXP (LOAD EXPONENT), DOUBLE MODE
                        ;       000000 --> 040252,125252,125252,125252 ==> 040052,125252,125252,125252
                        ;       FPS = 047600,   SRC = M2-R7,    AC = AC0
                        ;*****************************************************************************

004402  104400                    SCOPE
004404  000404                    BR      TST34

004406  040252  125252  125252  DAT34:  040252,125252,125252,125252
004414  125252

004416  170127  047617    TST34:  LDFPS   #047617         ;LOAD FLOATING POINT STATUS
004422  172467  177760            LDD     DAT34,  AC0     ;LOAD 040252,125252,125252,125252 INTO AC0
004426  176427  000000    FPI34:  LDEXP   #000000,AC0     ;LOAD 000000 INTO THE EXPONENT OF AC0
004432  170200                    STFPS   FPS             ;STORE FLOATING POINT STATUS
004434  022700  047600            CMP     #047600,FPS     ;CHECK FLOATING POINT STATUS
004440  001401                    BEQ     .+4             ;BRANCH IF OK
004442  104000                    HLT                     ;FPS NOT EQUAL TO 047600

004444  174067  174332            STD     AC0,    ANS1    ;STORE AC0 INTO ANS1 THRU ANS4
004450  022767  040052  174324    CMP     #040052,ANS1    ;CHECK ANS1
004456  001401                    BEQ     .+4             ;BRANCH IF OK
004460  104004                    HLT+4                   ;ANS1 NOT EQUAL TO 040052
```

```
004462  022767  125252  174314          CMP     #125252,ANS2    ;CHECK ANS2
004470  001401                          BEQ     .+4             ;BRANCH IF OK
004472  104004                          HLT+4                   ;ANS2 NOT EQUAL TO 125252

004474  022767  125252  174304          CMP     #125252,ANS3    ;CHECK ANS3
004502  001401                          BEQ     .+4             ;BRANCH IF OK
004504  104004                          HLT+4                   ;ANS3 NOT EQUAL TO 125252

004506  022767  125252  174274          CMP     #125252,ANS4    ;CHECK ANS4
004514  001401                          BEQ     .+4             ;BRANCH IF OK
004516  104004                          HLT+4                   ;ANS4 NOT EQUAL TO 125252

                        ;*****************************************************************
                        ;TEST 35:       TEST LDEXP (LOAD EXPONENT), DOUBLE MODE
                        ;       000005 --> 137777,177777,177777,177777 ==> 141377,177777,177777,177777
                        ;       FPS = 047610,   SRC = M2-R7,    AC = AC3
                        ;*****************************************************************

004520  104400                          SCOPE
004522  000404                          BR      TST35

004524  137777  177777  177777  DAT35:  137777,177777,177777,177777
004532  177777

004534  170127  047617          TST35:  LDFPS   #047617         ;LOAD FLOATING POINT STATUS
004540  172767  177760                  LDD     DAT35,  AC3     ;LOAD 137777,177777,177777,177777 INTO AC3
004544  176727  000005          FPI35:  LDEXP   #000005,AC3     ;LOAD 000005 INTO THE EXPONENT OF AC3
004550  170200                          STFPS   FPS             ;STORE FLOATING POINT STATUS
004552  022700  047610                  CMP     #047610,FPS     ;CHECK FLOATING POINT STATUS
004556  001401                          BEQ     .+4             ;BRANCH IF OK
004560  104000                          HLT                     ;FPS NOT EQUAL TO 047610

004562  174367  174214                  STD     AC3,    ANS1    ;STORE AC3 INTO ANS1 THRU ANS4
004566  022767  141377  174206          CMP     #141377,ANS1    ;CHECK ANS1
004574  001401                          BEQ     .+4             ;BRANCH IF OK
004576  104004                          HLT+4                   ;ANS1 NOT EQUAL TO 141377

004600  022767  177777  174176          CMP     #177777,ANS2    ;CHECK ANS2
004606  001401                          BEQ     .+4             ;BRANCH IF OK
004610  104004                          HLT+4                   ;ANS2 NOT EQUAL TO 177777

004612  022767  177777  174166          CMP     #177777,ANS3    ;CHECK ANS3
004620  001401                          BEQ     .+4             ;BRANCH IF OK
004622  104004                          HLT+4                   ;ANS3 NOT EQUAL TO 177777

004624  022767  177777  174156          CMP     #177777,ANS4    ;CHECK ANS4
004632  001401                          BEQ     .+4             ;BRANCH IF OK
004634  104004                          HLT+4                   ;ANS4 NOT EQUAL TO 177777

                        ;*****************************************************************
                        ;TEST 36:       TEST LDEXP (LOAD EXPONENT), DOUBLE MODE
                        ;       177601 --> 061616,143434,052525,107070 ==> 000216,143434,052525,107070
                        ;       FPS = 047600,   SRC = M2-R7,    AC = AC2
                        ;*****************************************************************
```

```
        004636  104400                                  SCOPE
        004640  000404                                  BR      TST36

        004642  061616  143434  052525  DAT36:  061616,143434,052525,107070
        004650  107070

        004652  170127  047617          TST36:  LDFPS   #047617         ;LOAD FLOATING POINT STATUS
        004656  172667  177760                  LDD     DAT36,  AC2     ;LOAD 061616,143434,052525,107070 INTO AC2
        004662  176627  177601          FPI36:  LDEXP   #177601,AC2     ;LOAD 177601 INTO THE EXPONENT OF AC2
        004666  170200                          STFPS   FPS             ;STORE FLOATING POINT STATUS
        004670  022700  047600                  CMP     #047600,FPS     ;CHECK FLOATING POINT STATUS
        004674  001401                          BEQ     .+4             ;BRANCH IF OK
        004676  104000                          HLT                     ;FPS NOT EQUAL TO 047600

        004700  174267  174076                  STD     AC2,    ANS1    ;STORE AC2 INTO ANS1 THRU ANS4
        004704  022767  000216  174070          CMP     #000216,ANS1    ;CHECK ANS1
        004712  001401                          BEQ     .+4             ;BRANCH IF OK
        004714  104004                          HLT+4                   ;ANS1 NOT EQUAL TO 000216

        004716  022767  143434  174060          CMP     #143434,ANS2    ;CHECK ANS2
        004724  001401                          BEQ     .+4             ;BRANCH IF OK
        004726  104004                          HLT+4                   ;ANS2 NOT EQUAL TO 143434

        004730  022767  052525  174050          CMP     #052525,ANS3    ;CHECK ANS3
        004736  001401                          BEQ     .+4             ;BRANCH IF OK
        004740  104004                          HLT+4                   ;ANS3 NOT EQUAL TO 052525

        004742  022767  107070  174040          CMP     #107070,ANS4    ;CHECK ANS4
        004750  001401                          BEQ     .+4             ;BRANCH IF OK
        004752  104004                          HLT+4                   ;ANS4 NOT EQUAL TO 107070

                                        ;*****************************************************************
                                        ;TEST 37:       TEST LDEXP (LOAD EXPONENT), DOUBLE MODE
                                        ;               000177 --> 161616,161616,161616,161616 ==> 177616,161616,161616,161616
                                        ;               FPS = 047610,   SRC = M2-R7,    AC = AC1
                                        ;*****************************************************************

        004754  104400                          SCOPE
        004756  000404                          BR      TST37

        004760  161616  161616  161616  DAT37:  161616,161616,161616,161616
        004766  161616

        004770  170127  047617          TST37:  LDFPS   #047617         ;LOAD FLOATING POINT STATUS
        004774  172567  177760                  LDD     DAT37,  AC1     ;LOAD 161616,161616,161616,161616 INTO AC1
        005000  176527  000177          FPI37:  LDEXP   #000177,AC1     ;LOAD 000177 INTO THE EXPONENT OF AC1
        005004  170200                          STFPS   FPS             ;STORE FLOATING POINT STATUS
        005006  022700  047610                  CMP     #047610,FPS     ;CHECK FLOATING POINT STATUS
        005012  001401                          BEQ     .+4             ;BRANCH IF OK
        005014  104000                          HLT                     ;FPS NOT EQUAL TO 047610

        005016  174167  173760                  STD     AC1,    ANS1    ;STORE AC1 INTO ANS1 THRU ANS4
        005022  022767  177616  173752          CMP     #177616,ANS1    ;CHECK ANS1
        005030  001401                          BEQ     .+4             ;BRANCH IF OK
        005032  104004                          HLT+4                   ;ANS1 NOT EQUAL TO 177616
```

```
005034  022767  161616  173742          CMP     #161616,ANS2    ;CHECK ANS2
005042  001401                          BEQ     .+4             ;BRANCH IF OK
005044  104004                          HLT+4                   ;ANS2 NOT EQUAL TO 161616

005046  022767  161616  173732          CMP     #161616,ANS3    ;CHECK ANS3
005054  001401                          BEQ     .+4             ;BRANCH IF OK
005056  104004                          HLT+4                   ;ANS3 NOT EQUAL TO 161616

005060  022767  161616  173722          CMP     #161616,ANS4    ;CHECK ANS4
005066  001401                          BEQ     .+4             ;BRANCH IF OK
005070  104004                          HLT+4                   ;ANS4 NOT EQUAL TO 161616

                        ;*****************************************************************
                        ;TEST 40:       TEST LDEXP (LOAD EXPONENT), DOUBLE MODE
                        ;       177777 --> 040000,000020,000020,000020 ==> 037600,000020,000020,000020
                        ;       FPS = 047600,   SRC = M2-R7,    AC = AC0
                        ;*****************************************************************

005072  104400                          SCOPE
005074  000404                          BR      TST40

005076  040000  000020  000020  DAT40:  040000,000020,000020,000020
005104  000020

005106  170127  047617          TST40:  LDFPS   #047617         ;LOAD FLOATING POINT STATUS
005112  172467  177760                  LDD     DAT40,  AC0     ;LOAD 040000,000020,000020,000020 INTO AC0
005116  176427  177777          FP140:  LDEXP   #177777,AC0     ;LOAD 177777 INTO THE EXPONENT OF AC0
005122  170200                          STFPS   FPS             ;STORE FLOATING POINT STATUS
005124  022700  047600                  CMP     #047600,FPS     ;CHECK FLOATING POINT STATUS
005130  001401                          BEQ     .+4             ;BRANCH IF OK
005132  104000                          HLT                     ;FPS NOT EQUAL TO 047600

005134  174067  173642                  STD     AC0,    ANS1    ;STORE AC0 INTO ANS1 THRU ANS4
005140  022767  037600  173634          CMP     #037600,ANS1    ;CHECK ANS1
005146  001401                          BEQ     .+4             ;BRANCH IF OK
005150  104004                          HLT+4                   ;ANS1 NOT EQUAL TO 037600

005152  022767  000020  173624          CMP     #000020,ANS2    ;CHECK ANS2
005160  001401                          BEQ     .+4             ;BRANCH IF OK
005162  104004                          HLT+4                   ;ANS2 NOT EQUAL TO 000020

005164  022767  000020  173614          CMP     #000020,ANS3    ;CHECK ANS3
005172  001401                          BEQ     .+4             ;BRANCH IF OK
005174  104004                          HLT+4                   ;ANS3 NOT EQUAL TO 000020

005176  022767  000020  173604          CMP     #000020,ANS4    ;CHECK ANS4
005204  001401                          BEQ     .+4             ;BRANCH IF OK
005206  104004                          HLT+4                   ;ANS4 NOT EQUAL TO 000020

                        ;*****************************************************************
                        ;TEST 41:       TEST LDEXP (LOAD EXPONENT), DOUBLE MODE
                        ;       177706 --> 125252,125252,125252,125252 ==> 121452,125252,125252,125252
                        ;       FPS = 047610,   SRC = M2-R7,    AC = AC3
                        ;*****************************************************************

005210  104400                          SCOPE
```

```
005212  000404                          BR      TST41

005214  125252  125252  125252  DAT41:  125252,125252,125252,125252
005222  125252

005224  170127  047617          TST41:  LDFPS   #047617         ;LOAD FLOATING POINT STATUS
005230  172767  177760                  LDD     DAT41,  AC3     ;LOAD 125252,125252,125252,125252 INTO AC3
005234  176727  177706          FPI41:  LDEXP   #177706,AC3     ;LOAD 177706 INTO THE EXPONENT OF AC3
005240  170200                          STFPS   FPS             ;STORE FLOATING POINT STATUS
005242  022700  047610                  CMP     #047610,FPS     ;CHECK FLOATING POINT STATUS
005246  001401                          BEQ     .+4             ;BRANCH IF OK
005250  104000                          HLT                     ;FPS NOT EQUAL TO 047610

005252  174367  173524                  STD     AC3,    ANS1    ;STORE AC3 INTO ANS1 THRU ANS4
005256  022767  121452  173516          CMP     #121452,ANS1    ;CHECK ANS1
005264  001401                          BEQ     .+4             ;BRANCH IF OK
005266  104004                          HLT+4                   ;ANS1 NOT EQUAL TO 121452

005270  022767  125252  173506          CMP     #125252,ANS2    ;CHECK ANS2
005276  001401                          BEQ     .+4             ;BRANCH IF OK
005300  104004                          HLT+4                   ;ANS2 NOT EQUAL TO 125252

005302  022767  125252  173476          CMP     #125252,ANS3    ;CHECK ANS3
005310  001401                          BEQ     .+4             ;BRANCH IF OK
005312  104004                          HLT+4                   ;ANS3 NOT EQUAL TO 125252

005314  022767  125252  173466          CMP     #125252,ANS4    ;CHECK ANS4
005322  001401                          BEQ     .+4             ;BRANCH IF OK
005324  104004                          HLT+4                   ;ANS4 NOT EQUAL TO 125252

                        ;*****************************************************************
                        ;TEST 42:       TEST LDEXP (LOAD EXPONENT), DOUBLE MODE
                        ;       000135 --> 052525,052525,052525,052525 ==> 067325,052525,052525,052525
                        ;       FPS = 047600,   SRC = M2-R7,    AC = AC2
                        ;*****************************************************************

005326  104400                          SCOPE
005330  000404                          BR      TST42

005332  052525  052525  052525  DAT42:  052525,052525,052525,052525
005340  052525

005342  170127  047617          TST42:  LDFPS   #047617         ;LOAD FLOATING POINT STATUS
005346  172667  177760                  LDD     DAT42,  AC2     ;LOAD 052525,052525,052525,052525 INTO AC2
005352  176627  000135          FPI42:  LDEXP   #000135,AC2     ;LOAD 000135 INTO THE EXPONENT OF AC2
005356  170200                          STFPS   FPS             ;STORE FLOATING POINT STATUS
005360  022700  047600                  CMP     #047600,FPS     ;CHECK FLOATING POINT STATUS
005364  001401                          BEQ     .+4             ;BRANCH IF OK
005366  104000                          HLT                     ;FPS NOT EQUAL TO 047600

005370  174267  173406                  STD     AC2,    ANS1    ;STORE AC2 INTO ANS1 THRU ANS4
005374  022767  067325  173400          CMP     #067325,ANS1    ;CHECK ANS1
005402  001401                          BEQ     .+4             ;BRANCH IF OK
005404  104004                          HLT+4                   ;ANS1 NOT EQUAL TO 067325

005406  022767  052525  173370          CMP     #052525,ANS2    ;CHECK ANS2
```

```
005414  001401                          BEQ     .+4.            ;BRANCH IF OK
005416  104004                          HLT+4                   ;ANS2 NOT EQUAL TO 052525

005420  022767  052525  173360          CMP     #052525,ANS3    ;CHECK ANS3
005426  001401                          BEQ     .+4             ;BRANCH IF OK
005430  104004                          HLT+4                   ;ANS3 NOT EQUAL TO 052525

005432  022767  052525  173350          CMP     #052525,ANS4    ;CHECK ANS4
005440  001401                          BEQ     .+4             ;BRANCH IF OK
005442  104004                          HLT+4                   ;ANS4 NOT EQUAL TO 052525

                        ;*****************************************************************
                        ;TEST 43:       TEST LDEXP (LOAD EXPONENT), DOUBLE MODE
                        ;       000200 --> 140425,125252,125252,125252 ==> 100025,125252,125252,125252
                        ;       FPS = 147616,   SRC = M2-R7,    AC = AC1
                        ;       FEC = 10,       FEA = FPI43
                        ;*****************************************************************

005444  104400                          SCOPE
005446  000404                          BR      TST43

005450  140425  125252  125252  DAT43:  140425,125252,125252,125252
005456  125252

005460  170127  047617          TST43:  LDFPS   #047617         ;LOAD FLOATING POINT STATUS
005464  172567  177760                  LDD     DAT43, AC1      ;LOAD 140425,125252,125252,125252 INTO AC1
005470  176527  000200          FPI43:  LDEXP   #000200,AC1     ;LOAD 000200 INTO THE EXPONENT OF AC1
005474  170200                          STFPS   FPS             ;STORE FLOATING POINT STATUS
005476  170367  173320                  STST    FEC             ;STORE EXCEPTION CODES
005502  022700  147616                  CMP     #147616,FPS     ;CHECK FLOATING POINT STATUS
005506  001401                          BEQ     .+4             ;BRANCH IF OK
005510  104000                          HLT                     ;FPS NOT EQUAL TO 147616

005512  022767  000010  173302          CMP     #10.    FEC     ;CHECK FLOATING EXCEPTION CODE
005520  001401                          BEQ     .+4             ;BRANCH IF OK
005522  104000                          HLT                     ;FEC NOT EQUAL TO 10

005524  022767  005470  173272          CMP     #FPI43, FEA     ;CHECK FLOATING EXCEPTION ADDRESS
005532  001401                          BEQ     .+4             ;BRANCH IF OK
005534  104000                          HLT                     ;FEA NOT EQUAL TO FPI43

005536  174167  173240                  STD     AC1,    ANS1    ;STORE AC1 INTO ANS1 THRU ANS4
005542  022767  100025  173232          CMP     #100025,ANS1    ;CHECK ANS1
005550  001401                          BEQ     .+4             ;BRANCH IF OK
005552  104004                          HLT+4                   ;ANS1 NOT EQUAL TO 100025

005554  022767  125252  173222          CMP     #125252,ANS2    ;CHECK ANS2
005562  001401                          BEQ     .+4             ;BRANCH IF OK
005564  104004                          HLT+4                   ;ANS2 NOT EQUAL TO 125252

005566  022767  125252  173212          CMP     #125252,ANS3    ;CHECK ANS3
005574  001401                          BEQ     .+4             ;BRANCH IF OK
005576  104004                          HLT+4                   ;ANS3 NOT EQUAL TO 125252

005600  022767  125252  173202          CMP     #125252,ANS4    ;CHECK ANS4
005606  001401                          BEQ     .+4             ;BRANCH IF OK
```

```
005610  104004                          HLT+4                   ;ANS4 NOT EQUAL TO 125252

                        ;*****************************************************************************
                        ;TEST 44:       TEST LDEXP (LOAD EXPONENT), DOUBLE MODE
                        ;       177600 --> 040525,125252,125252,125252 ==> 000125,125252,125252,125252
                        ;       FPS = 147604,   SRC = M2-R7,    AC = AC0
                        ;       FEC = 12,       FEA = FPI44
                        ;*****************************************************************************

005612  104400                          SCOPE
005614  000404                          BR      TST44

005616  040525  125252  125252  DAT44:  040525,125252,125252,125252
005624  125252

005626  170127  047617          TST44:  LDFPS   #047617         ;LOAD FLOATING POINT STATUS
005632  172467  177760                  LDD     DAT44,  AC0     ;LOAD 040525,125252,125252,125252 INTO AC0
005636  176427  177600          FPI44:  LDEXP   #177600,AC0     ;LOAD 177600 INTO THE EXPONENT OF AC0
005642  170200                          STFPS   FPS             ;STORE FLOATING POINT STATUS
005644  170367  173152                  STST    FEC             ;STORE EXCEPTION CODES
005650  022700  147604                  CMP     #147604,FPS     ;CHECK FLOATING POINT STATUS
005654  001401                          BEQ     .+4             ;BRANCH IF OK
005656  104000                          HLT                     ;FPS NOT EQUAL TO 147604

005660  022767  000012  173134          CMP     #12.    FEC     ;CHECK FLOATING EXCEPTION CODE
005666  001401                          BEQ     .+4             ;BRANCH IF OK
005670  104000                          HLT                     ;FEC NOT EQUAL TO 12

005672  022767  005636  173124          CMP     #FPI44, FEA     ;CHECK FLOATING EXCEPTION ADDRESS
005700  001401                          BEQ     .+4             ;BRANCH IF OK
005702  104000                          HLT                     ;FEA NOT EQUAL TO FPI44

005704  174067  173072                  STD     AC0,    ANS1    ;STORE AC0 INTO ANS1 THRU ANS4
005710  022767  000125  173064          CMP     #000125,ANS1    ;CHECK ANS1
005716  001401                          BEQ     .+4             ;BRANCH IF OK
005720  104004                          HLT+4                   ;ANS1 NOT EQUAL TO 000125

005722  022767  125252  173054          CMP     #125252,ANS2    ;CHECK ANS2
005730  001401                          BEQ     .+4             ;BRANCH IF OK
005732  104004                          HLT+4                   ;ANS2 NOT EQUAL TO 125252

005734  022767  125252  173044          CMP     #125252,ANS3    ;CHECK ANS3
005742  001401                          BEQ     .+4             ;BRANCH IF OK
005744  104004                          HLT+4                   ;ANS3 NOT EQUAL TO 125252

005746  022767  125252  173034          CMP     #125252,ANS4    ;CHECK ANS4
005754  001401                          BEQ     .+4             ;BRANCH IF OK
005756  104004                          HLT+4                   ;ANS4 NOT EQUAL TO 125252

                        ;*****************************************************************************
                        ;TEST 45:       TEST LDEXP (LOAD EXPONENT), DOUBLE MODE
                        ;       177575 --> 050505,050505,050505,050505 ==> 077305,050505,050505,050505
                        ;       FPS = 147600,   SRC = M2-R7,    AC = AC3
                        ;       FEC = 12,       FEA = FPI45
                        ;*****************************************************************************
```

```
005760  104400                          SCOPE
005762  000404                          BR      TST45

005764  050505  050505  050505  DAT45:  050505,050505,050505,050505
005772  050505

005774  170127  047617          TST45:  LDFPS   #047617         ;LOAD FLOATING POINT STATUS
006000  172767  177760                  LDD     DAT45,  AC3     ;LOAD 050505,050505,050505,050505 INTO AC3
006004  176727  177575          FP145:  LDEXP   #177575,AC3     ;LOAD 177575 INTO THE EXPONENT OF AC3
006010  170200                          STFPS   FPS             ;STORE FLOATING POINT STATUS
006012  170367  173004                  STST    FEC             ;STORE EXCEPTION CODES
006016  022700  147600                  CMP     #147600,FPS     ;CHECK FLOATING POINT STATUS
006022  001401                          BEQ     .+4             ;BRANCH IF OK
006024  104000                          HLT                     ;FPS NOT EQUAL TO 147600

006026  022767  000012  172766          CMP     #12,    FEC     ;CHECK FLOATING EXCEPTION CODE
006034  001401                          BEQ     .+4             ;BRANCH IF OK
006036  104000                          HLT                     ;FEC NOT EQUAL TO 12

006040  022767  006004  172756          CMP     #FP145, FEA     ;CHECK FLOATING EXCEPTION ADDRESS
006046  001401                          BEQ     .+4             ;BRANCH IF OK
006050  104000                          HLT                     ;FEA NOT EQUAL TO FP145

006052  174367  172724                  STD     AC3,    ANS1    ;STORE AC3 INTO ANS1 THRU ANS4
006056  022767  077305  172716          CMP     #077305,ANS1    ;CHECK ANS1
006064  001401                          BEQ     .+4             ;BRANCH IF OK
006066  104004                          HLT+4                   ;ANS1 NOT EQUAL TO 077305

006070  022767  050505  172706          CMP     #050505,ANS2    ;CHECK ANS2
006076  001401                          BEQ     .+4             ;BRANCH IF OK
006100  104004                          HLT+4                   ;ANS2 NOT EQUAL TO 050505

006102  022767  050505  172676          CMP     #050505,ANS3    ;CHECK ANS3
006110  001401                          BEQ     .+4             ;BRANCH IF OK
006112  104004                          HLT+4                   ;ANS3 NOT EQUAL TO 050505

006114  022767  050505  172666          CMP     #050505,ANS4    ;CHECK ANS4
006122  001401                          BEQ     .+4             ;BRANCH IF OK
006124  104004                          HLT+4                   ;ANS4 NOT EQUAL TO 050505

                        ;*****************************************************************
                        ;TEST 46:       TEST LDEXP (LOAD EXPONENT), DOUBLE MODE
                        ;       000203 --> 140425,125252,125252,125252 ==> 100625,125252,125252,125252
                        ;       FPS = 046612,   SRC = M2-R7,    AC = AC2
                        ;*****************************************************************

006126  104400                          SCOPE
006130  000404                          BR      TST46

006132  140425  125252  125252  DAT46:  140425,125252,125252,125252
006140  125252

006142  170127  046605          TST46:  LDFPS   #046605         ;LOAD FLOATING POINT STATUS
006146  172667  177760                  LDD     DAT46,  AC2     ;LOAD 140425,125252,125252,125252 INTO AC2
006152  176627  000203          FP146:  LDEXP   #000203,AC2     ;LOAD 000203 INTO THE EXPONENT OF AC2
006156  170200                          STFPS   FPS             ;STORE FLOATING POINT STATUS
```

```
006160  022700  046612          CMP     #046612,FPS     ;CHECK FLOATING POINT STATUS
006164  001401                  BEQ     .+4             ;BRANCH IF OK
006166  104000                  HLT                     ;FPS NOT EQUAL TO 046612

006170  174267  172606          STD     AC2,    ANS1    ;STORE AC2 INTO ANS1 THRU ANS4
006174  022767  100625  172600  CMP     #100625,ANS1    ;CHECK ANS1
006202  001401                  BEQ     .+4             ;BRANCH IF OK
006204  104004                  HLT+4                   ;ANS1 NOT EQUAL TO 100625

006206  022767  125252  172570  CMP     #125252,ANS2    ;CHECK ANS2
006214  001401                  BEQ     .+4             ;BRANCH IF OK
006216  104004                  HLT+4                   ;ANS2 NOT EQUAL TO 125252

006220  022767  125252  172560  CMP     #125252,ANS3    ;CHECK ANS3
006226  001401                  BEQ     .+4             ;BRANCH IF OK
006230  104004                  HLT+4                   ;ANS3 NOT EQUAL TO 125252

006232  022767  125252  172550  CMP     #125252,ANS4    ;CHECK ANS4
006240  001401                  BEQ     .+4             ;BRANCH IF OK
006242  104004                  HLT+4                   ;ANS4 NOT EQUAL TO 125252

                        ;*****************************************************************
                        ;TEST 47:       TEST LDEXP (LOAD EXPONENT), DOUBLE MODE
                        ;       177577 --> 040525,125252,125252,125252 ==> 000000,000000,000000,000000
                        ;       FPS = 045604,   SRC = M2-R7,    AC = AC0
                        ;*****************************************************************

006244  104400                  SCOPE
006246  000404                  BR      TST47

006250  040525  125252  125252  DAT47:  040525,125252,125252,125252
006256  125252

006260  170127  045617          TST47:  LDFPS   #045617         ;LOAD FLOATING POINT STATUS
006264  172467  177760                  LDD     DAT47,  AC0     ;LOAD 040525,125252,125252,125252 INTO AC0
006270  176427  177577          FP247:  LDEXP   #177577,AC0     ;LOAD 177577 INTO THE EXPONENT OF AC0
006274  170200                  STFPS   FPS                     ;STORE FLOATING POINT STATUS
006276  022700  045604          CMP     #045604,FPS             ;CHECK FLOATING POINT STATUS
006302  001401                  BEQ     .+4                     ;BRANCH IF OK
006304  104000                  HLT                             ;FPS NOT EQUAL TO 045604

006306  174067  172470          STD     AC0,    ANS1            ;STORE AC0 INTO ANS1 THRU ANS4
006312  022767  000000  172462  CMP     #000000,ANS1            ;CHECK ANS1
006320  001401                  BEQ     .+4                     ;BRANCH IF OK
006322  104004                  HLT+4                           ;ANS1 NOT EQUAL TO 000000

006324  022767  000000  172452  CMP     #000000,ANS2            ;CHECK ANS2
006332  001401                  BEQ     .+4                     ;BRANCH IF OK
006334  104004                  HLT+4                           ;ANS2 NOT EQUAL TO 000000

006336  022767  000000  172442  CMP     #000000,ANS3            ;CHECK ANS3
006344  001401                  BEQ     .+4                     ;BRANCH IF OK
006346  104004                  HLT+4                           ;ANS3 NOT EQUAL TO 000000

006350  022767  000000  172432  CMP     #000000,ANS4            ;CHECK ANS4
006356  001401                  BEQ     .+4                     ;BRANCH IF OK
```

```
006360  104004                          HLT+4                   ;ANS4 NOT EQUAL TO 000000

                        ;*************************************************************************
                        ;TEST 50:       TEST LDEXP (LOAD EXPONENT), DOUBLE MODE
                        ;       177706 --> 156212,121212,121212,121212 ==> 121412,121212,121212,121212
                        ;       FPS = 047610,   SRC = M0-R5,    AC = AC0
                        ;*************************************************************************

006362  104400                          SCOPE
006364  000404                          BR      TST50

006366  156212  121212  121212  DAT50:  156212,121212,121212,121212
006374  121212

006376  170127  047617          TST50:  LDFPS   #047617         ;LOAD FLOATING POINT STATUS
006402  172467  177760                  LDD     DAT50,  AC0     ;LOAD 156212,121212,121212,121212 INTO AC0
006406  012705  177706                  MOV     #177706,R5      ;PUT EXPONENT INTO R5
006412  176405                  FPI50:  LDEXP   R5,     AC0     ;LOAD 177706 INTO THE EXPONENT OF AC0
006414  170200                          STFPS   FPS             ;STORE FLOATING POINT STATUS
006416  022700  047610                  CMP     #047610,FPS     ;CHECK FLOATING POINT STATUS
006422  001401                          BEQ     .+4             ;BRANCH IF OK
006424  104000                          HLT                     ;FPS NOT EQUAL TO 047610

006426  174067  172350                  STD     AC0,    ANS1    ;STORE AC0 INTO ANS1 THRU ANS4
006432  022767  121412  172342          CMP     #121412,ANS1    ;CHECK ANS1
006440  001401                          BEQ     .+4             ;BRANCH IF OK
006442  104004                          HLT+4                   ;ANS1 NOT EQUAL TO 121412

006444  022767  121212  172332          CMP     #121212,ANS2    ;CHECK ANS2
006452  001401                          BEQ     .+4             ;BRANCH IF OK
006454  104004                          HLT+4                   ;ANS2 NOT EQUAL TO 121212

006456  022767  121212  172322          CMP     #121212,ANS3    ;CHECK ANS3
006464  001401                          BEQ     .+4             ;BRANCH IF OK
006466  104004                          HLT+4                   ;ANS3 NOT EQUAL TO 121212

006470  022767  121212  172312          CMP     #121212,ANS4    ;CHECK ANS4
006476  001401                          BEQ     .+4             ;BRANCH IF OK
006500  104004                          HLT+4                   ;ANS4 NOT EQUAL TO 121212

                        ;*************************************************************************
                        ;TEST 51:       TEST LDEXP (LOAD EXPONENT), DOUBLE MODE
                        ;       000135 --> 052525,052525,052525,052525 ==> 067325,052525,052525,052525
                        ;       FPS = 047600,   SRC = M0-R2,    AC = AC2
                        ;*************************************************************************

006502  104400                          SCOPE
006504  000404                          BR      TST51

006506  052525  052525  052525  DAT51:  052525,052525,052525,052525
006514  052525

006516  170127  047617          TST51:  LDFPS   #047617         ;LOAD FLOATING POINT STATUS
006522  172667  177760                  LDD     DAT51,  AC2     ;LOAD 052525,052525,052525,052525 INTO AC2
006526  012702  000135                  MOV     #000135,R2      ;PUT EXPONENT INTO R2
006532  176602                  FPI51:  LDEXP   R2,     AC2     ;LOAD 000135 INTO THE EXPONENT OF AC2
```

```
006534  170200                          STFPS   FPS             ;STORE FLOATING POINT STATUS
006536  022700  047600                  CMP     #047600,FPS     ;CHECK FLOATING POINT STATUS
006542  001401                          BEQ     .+4             ;BRANCH IF OK
006544  104000                          HLT                     ;FPS NOT EQUAL TO 04760C

006546  174267  172230                  STD     AC2,    ANS1    ;STORE AC2 INTO ANS1 THRU ANS4
006552  022767  067325  172222          CMP     #067325,ANS1    ;CHECK ANS1
006560  001401                          BEQ     .+4             ;BRANCH IF OK
006562  104004                          HLT+4                   ;ANS1 NOT EQUAL TO 067325

006564  022767  052525  172212          CMP     #052525,ANS2    ;CHECK ANS2
006572  001401                          BEQ     .+4             ;BRANCH IF OK
006574  104004                          HLT+4                   ;ANS2 NOT EQUAL TO 052525

006576  022767  052525  172202          CMP     #052525,ANS3    ;CHECK ANS3
006604  001401                          BEQ     .+4             ;BRANCH IF OK
006606  104004                          HLT+4                   ;ANS3 NOT EQUAL TO 052525

006610  022767  052525  172172          CMP     #052525,ANS4    ;CHECK ANS4
006616  001401                          BEQ     .+4             ;BRANCH IF OK
006620  104004                          HLT+4                   ;ANS4 NOT EQUAL TO 052525

                        ;*****************************************************************
                        ;TEST 52:       TEST STEXP (STORE EXPONENT), DOUBLE MODE
                        ;       000000,000000,000000,000000 --> 177600
                        ;       FPS = 047610,   FCC = 10,       AC = AC1,       DST = M6-R7
                        ;*****************************************************************

006622  104400                          SCOPE
006624  000404                          BR      TST52

006626  000000  000000  000000  DAT52:  000000,000000,000000,000000
006634  000000

006636  170127  047617          TST52:  LDFPS   #047617         ;LOAD FLOATING POINT STATUS
006642  172567  177760                  LDD     DAT52,  AC1     ;LOAD 000000,000000,000000,000000 INTO AC1
006646  175167  172130                  STEXP   AC1,    ANS1    ;STORE THE EXPONENT OF AC1 IN ANS1
006652  013767  177776  172124          MOV     @#PS,   ANS2    ;GET CPU STATUS
006660  042767  177760  172116          BIC     #177760,ANS2    ;SAVE CONDITION CODES
006666  170200                          STFPS   FPS             ;STORE FLOATING POINT STATUS
006670  022700  047610                  CMP     #047610,FPS     ;CHECK FLOATING POINT STATUS
006674  001401                          BEQ     .+4             ;BRANCH IF OK
006676  104000                          HLT                     ;FPS NOT EQUAL TO 047610

006700  022767  177600  172074          CMP     #177600,ANS1    ;CHECK ANS1
006706  001401                          BEQ     .+4             ;BRANCH IF OK
006710  104001                          HLT+1                   ;ANS1 NOT EQUAL TO 177600

006712  022767  000010  172064          CMP     #10.    ANS2    ;CHECK ANS2
006720  001401                          BEQ     .+4             ;BRANCH IF OK
006722  104002                          HLT+2                   ;WRONG CONDITION CODES!

                        ;*****************************************************************
                        ;TEST 53:       TEST STEXP (STORE EXPONENT), DOUBLE MODE
                        ;       040161,061434,143434,125252 --> 000000
                        ;       FPS = 047604,   FCC = 04,       AC = AC3,       DST = M6-R7
```

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

006724  104400                        SCOPE
006726  000404                        BR      TST53

006730  040161  061434  143434  DAT53:  040161,061434,143434,125252
006736  125252

006740  170127  047617          TST53: LDFPS   #047617          ;LOAD FLOATING POINT STATUS
006744  172767  177760                 LDD     DAT53,  AC3     ;LOAD 040161,061434,143434,125252 INTO AC3
006750  175367  172026                 STEXP   AC3,    ANS1    ;STORE THE EXPONENT OF AC3 IN ANS1
006754  013767  177776  172022         MOV     @#PS,   ANS2    ;GET CPU STATUS
006762  042767  177760  172014         BIC     #177760,ANS2    ;SAVE CONDITION CODES
006770  170200                        STFPS   FPS             ;STORE FLOATING POINT STATUS
006772  022700  047604                CMP     #047604,FPS     ;CHECK FLOATING POINT STATUS
006776  001401                        BEQ     .+4             ;BRANCH IF OK
007000  104000                        HLT                     ;FPS NOT EQUAL TO 047604

007002  022767  000000  171772        CMP     #000000,ANS1    ;CHECK ANS1
007010  001401                        BEQ     .+4             ;BRANCH IF OK
007012  104001                        HLT+1                   ;ANS1 NOT EQUAL TO 000000

007014  022767  000004  171762        CMP     #04,    ANS2    ;CHECK ANS2
007022  001401                        BEQ     .+4             ;BRANCH IF OK
007024  104002                        HLT+2                   ;WRONG CONDITION CODES!

                        ;*****************************************************************************
                        ;TEST 54:       TEST STEXP (STORE EXPONENT), DOUBLE MODE
                        ;       040200,125250,125250,125252 --> 000001
                        ;       FPS = 047600,   FCC = 00,       AC = AC2,       DST = M6-R7
                        ;*****************************************************************************

007026  104400                        SCOPE
007030  000404                        BR      TST54

007032  040200  125250  125250  DAT54:  040200,125250,125250,125252
007040  125252

007042  170127  047617          TST54: LDFPS   #047617          ;LOAD FLOATING POINT STATUS
007046  172667  177760                 LDD     DAT54,  AC2     ;LOAD 040200,125250,125250,125252 INTO AC2
007052  175267  171724                 STEXP   AC2,    ANS1    ;STORE THE EXPONENT OF AC2 IN ANS1
007056  013767  177776  171720         MOV     @#PS,   ANS2    ;GET CPU STATUS
007064  042767  177760  171712         BIC     #177760,ANS2    ;SAVE CONDITION CODES
007072  170200                        STFPS   FPS             ;STORE FLOATING POINT STATUS
007074  022700  047600                CMP     #047600,FPS     ;CHECK FLOATING POINT STATUS
007100  001401                        BEQ     .+4             ;BRANCH IF OK
007102  104000                        HLT                     ;FPS NOT EQUAL TO 047600

007104  022767  000001  171670        CMP     #000001,ANS1    ;CHECK ANS1
007112  001401                        BEQ     .+4             ;BRANCH IF OK
007114  104001                        HLT+1                   ;ANS1 NOT EQUAL TO 000001

007116  022767  000000  171660        CMP     #00,    ANS2    ;CHECK ANS2
007124  001401                        BEQ     .+4             ;BRANCH IF OK
007126  104002                        HLT+2                   ;WRONG CONDITION CODES!
```

```
                        ;*************************************************************************
                        ;TEST 55:       TEST STEXP (STORE EXPONENT), DOUBLE MODE
                        ;       042525,052525,052525,052525 --> 000012
                        ;       FPS = 047600,   FCC = 00,       AC = AC3,       DST = M6-R7
                        ;*************************************************************************

007130  104400                  SCOPE
007132  000404                  BR      TST55

007134  042525  052525  052525  DAT55:  042525,052525,052525,052525
007142  052525

007144  170127  047617          TST55:  LDFPS   #047617         ;LOAD FLOATING POINT STATUS
007150  172767  177760                  LDD     DAT55,  AC3     ;LOAD 042525,052525,052525,052525 INTO AC3
007154  175367  171622                  STEXP   AC3,    ANS1    ;STORE THE EXPONENT OF AC3 IN ANS1
007160  013767  177776  171616          MOV     @#PS,   ANS2    ;GET CPU STATUS
007166  042767  177760  171610          BIC     #177760,ANS2    ;SAVE CONDITION CODES
007174  170200                  STFPS   FPS             ;STORE FLOATING POINT STATUS
007176  022700  047600                  CMP     #047600,FPS     ;CHECK FLOATING POINT STATUS
007202  001401                  BEQ     .+4             ;BRANCH IF OK
007204  104000                  HLT                     ;FPS NOT EQUAL TO 047600

007206  022767  000012  171566          CMP     #000012,ANS1    ;CHECK ANS1
007214  001401                  BEQ     .+4             ;BRANCH IF OK
007216  104001                  HLT+1                   ;ANS1 NOT EQUAL TO 000012

007220  022767  000000  171556          CMP     #00,    ANS2    ;CHECK ANS2
007226  001401                  BEQ     .+4             ;BRANCH IF OK
007230  104002                  HLT+2                   ;WRONG CONDITION CODES!

                        ;*************************************************************************
                        ;TEST 56:       TEST STEXP (STORE EXPONENT), DOUBLE MODE
                        ;       125252,125252,125252,125252 --> 177725
                        ;       FPS = 047610,   FCC = 10,       AC = AC1,       DST = M6-R7
                        ;*************************************************************************

007232  104400                  SCOPE
007234  000404                  BR      TST56

007236  125252  125252  125252  DAT56:  125252,125252,125252,125252
007244  125252

007246  170127  047617          TST56:  LDFPS   #047617         ;LOAD FLOATING POINT STATUS
007252  172567  177760                  LDD     DAT56,  AC1     ;LOAD 125252,125252,125252,125252 INTO AC1
007256  175167  171520                  STEXP   AC1,    ANS1    ;STORE THE EXPONENT OF AC1 IN ANS1
007262  013767  177776  171514          MOV     @#PS,   ANS2    ;GET CPU STATUS
007270  042767  177760  171506          BIC     #177760,ANS2    ;SAVE CONDITION CODES
007276  170200                  STFPS   FPS             ;STORE FLOATING POINT STATUS
007300  022700  047610                  CMP     #047610,FPS     ;CHECK FLOATING POINT STATUS
007304  001401                  BEQ     .+4             ;BRANCH IF OK
007306  104000                  HLT                     ;FPS NOT EQUAL TO 047610

007310  022767  177725  171464          CMP     #177725,ANS1    ;CHECK ANS1
007316  001401                  BEQ     .+4             ;BRANCH IF OK
007320  104001                  HLT+1                   ;ANS1 NOT EQUAL TO 177725
```

```
007322  022767  000010  171454          CMP     #10,    ANS2    ;CHECK ANS2
007330  001401                          BEQ     .+4             ;BRANCH IF OK
007332  104002                          HLT+2                   ;WRONG CONDITION CODES!

                        ;*****************************************************************
                        ;TEST 57:       TEST STEXP (STORE EXPONENT), DOUBLE MODE
                        ;       052525,052525,052525,052525 --> 000052
                        ;       FPS = 047600,   FCC = 00,       AC = AC3,       DST = M6-R7
                        ;*****************************************************************

007334  104400                          SCOPE
007336  000404                          BR      TST57

007340  052525  052525  052525  DAT57:  052525,052525,052525,052525
007346  052525

007350  170127  047617          TST57:  LDFPS   #047617         ;LOAD FLOATING POINT STATUS
007354  172767  177760                  LDD     DAT57,  AC3     ;LOAD 052525,052525,052525,052525 INTO AC3
007360  175367  171416                  STEXP   AC3,    ANS1    ;STORE THE EXPONENT OF AC3 IN ANS1
007364  013767  177776  171412          MOV     @#PS,   ANS2    ;GET CPU STATUS
007372  042767  177760  171404          BIC     #177760,ANS2    ;SAVE CONDITION CODES
007400  170200                          STFPS   FPS             ;STORE FLOATING POINT STATUS
007402  022700  047600                  CMP     #047600,FPS     ;CHECK FLOATING POINT STATUS
007406  001401                          BEQ     .+4             ;BRANCH IF OK
007410  104000                          HLT                     ;FPS NOT EQUAL TO 047600

007412  022767  000052  171362          CMP     #000052,ANS1    ;CHECK ANS1
007420  001401                          BEQ     .+4             ;BRANCH IF OK
007422  104001                          HLT+1                   ;ANS1 NOT EQUAL TO 000052

007424  022767  000000  171352          CMP     #00,    ANS2    ;CHECK ANS2
007432  001401                          BEQ     .+4             ;BRANCH IF OK
007434  104002                          HLT+2                   ;WRONG CONDITION CODES!

                        ;*****************************************************************
                        ;TEST 60:       TEST STEXP (STORE EXPONENT), DOUBLE MODE
                        ;       177777,177777,177777,177777 --> 000177
                        ;       FPS = 047600,   FCC = 00,       AC = AC1,       DST = M6-R7
                        ;*****************************************************************

007436  104400                          SCOPE
007440  000404                          BR      TST60

007442  177777  177777  177777  DAT60:  177777,177777,177777,177777
007450  177777

007452  170127  047617          TST60:  LDFPS   #047617         ;LOAD FLOATING POINT STATUS
007456  172567  177760                  LDD     DAT60,  AC1     ;LOAD 177777,177777,177777,177777 INTO AC1
007462  175167  171314                  STEXP   AC1,    ANS1    ;STORE THE EXPONENT OF AC1 IN ANS1
007466  013767  177776  171310          MOV     @#PS,   ANS2    ;GET CPU STATUS
007474  042767  177760  171302          BIC     #177760,ANS2    ;SAVE CONDITION CODES
007502  170200                          STFPS   FPS             ;STORE FLOATING POINT STATUS
007504  022700  047600                  CMP     #047600,FPS     ;CHECK FLOATING POINT STATUS
007510  001401                          BEQ     .+4             ;BRANCH IF OK
007512  104000                          HLT                     ;FPS NOT EQUAL TO 047600
```

```
007514  022767  000177  171260          CMP     #000177,ANS1    ;CHECK ANS1
007522  001401                          BEQ     .+4             ;BRANCH IF OK
007524  104001                          HLT+1                   ;ANS1 NOT EQUAL TO 000177

007526  022767  000000  171250          CMP     #00,    ANS2    ;CHECK ANS2
007534  001401                          BEQ     .+4             ;BRANCH IF OK
007536  104002                          HLT+2                   ;WRONG CONDITION CODES!

                        ;*****************************************************************
                        ;TEST 61:       TEST STEXP (STORE EXPONENT), DOUBLE MODE
                        ;       000200,000000,000000,000000 --> 177601
                        ;       FPS = 047610,   FCC = 10,       AC = AC3,       DST = M6-R7
                        ;*****************************************************************

007540  104400                          SCOPE
007542  000404                          BR      TST61

007544  000200  000000  000000  DAT61:  000200,000000,000000,000000
007552  000000

007554  170127  047617          TST61:  LDFPS   #047617         ;LOAD FLOATING POINT STATUS
007560  172767  177760                  LDD     DAT61,  AC3     ;LOAD 000200,000000,000000,000000 INTO AC3
007564  175367  171212                  STEXP   AC3,    ANS1    ;STORE THE EXPONENT OF AC3 IN ANS1
007570  013767  177776  171206          MOV     @#PS,   ANS2    ;GET CPU STATUS
007576  042767  177760  171200          BIC     #177760,ANS2    ;SAVE CONDITION CODES
007604  170200                          STFPS   FPS             ;STORE FLOATING POINT STATUS
007606  022700  047610                  CMP     #047610,FPS     ;CHECK FLOATING POINT STATUS
007612  001401                          BEQ     .+4             ;BRANCH IF OK
007614  104000                          HLT                     ;FPS NOT EQUAL TO 047610

007616  022767  177601  171156          CMP     #177601,ANS1    ;CHECK ANS1
007624  001401                          BEQ     .+4             ;BRANCH IF OK
007626  104001                          HLT+1                   ;ANS1 NOT EQUAL TO 177601

007630  022767  000010  171146          CMP     #10.    ANS2    ;CHECK ANS2
007636  001401                          BEQ     .+4             ;BRANCH IF OK
007640  104002                          HLT+2                   ;WRONG CONDITION CODES!

                        ;*****************************************************************
                        ;TEST 62:       TEST STEXP (STORE EXPONENT), DOUBLE MODE
                        ;       125252,125252,125252,125252 --> 177725
                        ;       FPS = 047610,   FCC = 10,       AC = AC0,       DST = M0-R3
                        ;*****************************************************************

007642  104400                          SCOPE
007644  000404                          BR      TST62

007646  125252  125252  125252  DAT62:  125252,125252,125252,125252
007654  125252

007656  170127  047617          TST62:  LDFPS   #047617         ;LOAD FLOATING POINT STATUS
007662  172467  177760                  LDD     DAT62,  AC0     ;LOAD 125252,125252,125252,125252 INTO AC0
007666  175003                          STEXP   AC0,    R3      ;STORE THE EXPONENT OF AC0 IN R3
007670  013767  177776  171106          MOV     @#PS,   ANS2    ;GET CPU STATUS
007676  042767  177760  171100          BIC     #177760,ANS2    ;SAVE CONDITION CODES
007704  010367  171072                  MOV     R3,     ANS1    ;SAVE R3 FOR TYPING
```

```
                                                    ;IF NOT EQUAL TO 047600
                                                    ;BRANCH IF OK
                                                    ;FPS NOT EQUAL TO 047600

                                                    ;CHECK ANS1
                                                    ;BRANCH IF OK
                                                    ;ANS1 NOT EQUAL TO 177735

                                                    ;CHECK ANS2
                                                    ;BRANCH IF OK
                                                    ;WRONG CONDITION CODES!

;*****************************************************************************
;TEST 63:       TEST STEXP (STORE EXPONENT), DOUBLE MODE
;       052525,052525,052525,052525 --> 000052
;       FPS = 047600,   FCC = 00,       AC = AC1,       CST = MC-BR
;*****************************************************************************

                        SCOPE
                        BR      TST63

DAT63:  052525,052525,052525,052525

TST63:  LDFPS   #047617         ;LOAD FLOATING POINT STATUS
        LDD     DAT63,  AC1     ;LOAD 052525,052525,052525,052525 INTO AC1
        STEXP   AC1,    R4      ;STORE THE EXPONENT OF AC1 IN R4
        MOV     #PS,    ANS2    ;GET CPU STATUS
        BIC     #177760,ANS2    ;SAVE CONDITION CODES
        MOV     R4,     ANS1    ;SAVE R4 FOR TYPING
        STFPS   FPS             ;STORE FLOATING POINT STATUS
        CMP     #047600,FPS     ;CHECK FLOATING POINT STATUS
        BEQ     .+4             ;BRANCH IF OK
        HLT                     ;FPS NOT EQUAL TO 047600

        CMP     #000052,ANS1    ;CHECK ANS1
        BEQ     .+4             ;BRANCH IF OK
        HLT+1                   ;ANS1 NOT EQUAL TO 000052

        CMP     #00,    ANS2    ;CHECK ANS2
        BEQ     .+4             ;BRANCH IF OK
        HLT+2                   ;WRONG CONDITION CODES!

;*****************************************************************************
;TEST 64:       TEST STEXP (STORE EXPONENT), DOUBLE MODE
;       050252,125252,125252,052525 --> 000041
;       FPS = 047600,   FCC = 00,       AC = AC2,       CST = MC-BR
;*****************************************************************************

        SCOPE
        BR      TST64

DAT64:  050252,125252,125252,052525
```

```
010066  170127  047617          TST64:  LDFPS   #047617         ;LOAD FLOATING POINT STATUS
010072  172667  177760                  LDD     DAT64,  AC2     ;LOAD 050252,125252,125252,052525 INTO AC2
010076  005067  000002                  CLR     ANR64
010102  175227                          STEXP   AC2,    (PC)+   ;STORE THE EXPONENT OF AC2 IN .+2
010104  000000                  ANR64:  0
010106  000403                          BR      .+10
010110  000000                          HALT
010112  000000                          HALT
010114  000000                          HALT
010116  013767  177776  170660          MOV     @#PS,   ANS2    ;GET CPU STATUS
010124  042767  177760  170652          BIC     #177760,ANS2    ;SAVE CONDITION CODES
010132  170200                          STFPS   FPS             ;STORE FLOATING POINT STATUS
010134  022700  047600                  CMP     #047600,FPS     ;CHECK FLOATING POINT STATUS
010140  001401                          BEQ     .+4             ;BRANCH IF OK
010142  104000                          HLT                     ;FPS NOT EQUAL TO 047600

010144  016767  177734  170630          MOV     ANR64,  ANS1
010152  022767  000041  170622          CMP     #000041,ANS1    ;CHECK ANS1
010160  001401                          BEQ     .+4             ;BRANCH IF OK
010162  104001                          HLT+1                   ;ANS1 NOT EQUAL TO 000041

010164  022767  000000  170612          CMP     #00,    ANS2    ;CHECK ANS2
010172  001401                          BEQ     .+4             ;BRANCH IF OK
010174  104002                          HLT+2                   ;WRONG CONDITION CODES!
```

```
;*****************************************************************
;TEST 65:       TEST STEXP (STORE EXPONENT), DOUBLE MODE
;       027616,161616,034343,070707 --> 177737
;       FPS = 047610,   FCC = 10,       AC = AC0,       DST = M2-R7
;*****************************************************************

010176  104400                          SCOPE
010200  000404                          BR      TST65

010202  027616  161616  034343  DAT65:  027616,161616,034343,070707
010210  070707

010212  170127  047617          TST65:  LDFPS   #047617         ;LOAD FLOATING POINT STATUS
010216  172467  177760                  LDD     DAT65,  AC0     ;LOAD 027616,161616,034343,070707 INTO AC0
010222  005067  000002                  CLR     ANR65
010226  175027                          STEXP   AC0,    (PC)+   ;STORE THE EXPONENT OF AC0 IN .+2
010230  000000                  ANR65:  0
010232  000403                          BR      .+10
010234  000000                          HALT
010236  000000                          HALT
010240  000000                          HALT
010242  013767  177776  170534          MOV     @#PS,   ANS2    ;GET CPU STATUS
010250  042767  177760  170526          BIC     #177760,ANS2    ;SAVE CONDITION CODES
010256  170200                          STFPS   FPS             ;STORE FLOATING POINT STATUS
010260  022700  047610                  CMP     #047610,FPS     ;CHECK FLOATING POINT STATUS
010264  001401                          BEQ     .+4             ;BRANCH IF OK
010266  104000                          HLT                     ;FPS NOT EQUAL TO 047610

010270  016767  177734  170504          MOV     ANR65,  ANS1
010276  022767  177737  170476          CMP     #177737,ANS1    ;CHECK ANS1
010304  001401                          BEQ     .+4             ;BRANCH IF OK
```

```
010306  104001                          HLT+1                   ;ANS1 NOT EQUAL TO 177737

010310  022767  000010  177466          CMP     #10.,   ANS2    ;CHECK ANS2
010316  001401                          BEQ     .+4             ;BRANCH IF OK
010320  104002                          HLT+2                   ;WRONG CONDITION CODES!
```

```
010322  104400                  DONE:   SCOPE
010324  032737  002000  177570          BIT     #SW10,@#SWR     ;PING THE BELL?
010332  001005                          BNE     1$              ;NO!
010334  012767  000207  001242          MOV     #BELL,.TYPE     ;TYPE A BELL
010342  000004  011604                  TYPE    ,.TYPE
010346  005046                  1$:     CLR     -(6)            ;CLEAR TRACE TRAP
010350  032737  010000  177570          BIT     #SW12,@#SWR     ;RUN WITH TRT?
010356  001010                          BNE     2$
010360  005167  001222                  COM     TRPB
010364  100005                          BPL     2$
010366  052716  000020                  BIS     #20,(6)         ;SET TRACE TRAP
010372  012746  001062                  MOV     #BEGIN,-(6)     ;JUMP TO START OF TEST
010375  000412                          BR      YESRT
010400  012746  001062          2$:     MOV     #BEGIN,-(6)     ;JUMP TO START OF TEST
010404  013700  000042                  MOV     @#42,R0         ;GET MONITOR ADDRESS
010410  001404                          BEQ     3$              ;IF NONE
010412  004710                          JSR     7,,0,           ;GO TO MONITOR
010414  000240                          NOP
010416  000240                          NOP
010420  000240                          NOP
010422  000002                  3$:     RTI
010424  000002                  YESRT:  RTI                     ;RETURN TO PROGRAM FROM TRAP

010426  032737  000400  177570  .EMT:   BIT     #SW08,@#SWR     ;KILL LDUB OR LOOP ON SPEC. TEST
010434  001404                          BEQ     1$
010436  123767  177570  170334          CMPB    @#SWR,ICNT      ;ON RIGHT TEST?   #SW7-0#
010444  001437                          BEQ     OVER
010446  113703  177570          1$:     MOVB    @#SWR,R3        ;GET UB BITS
010452  170003                          LDUB
010454  032737  040000  177570          BIT     #SW14,@#SWR     ;LOOP ON TEST
010462  001026                          BNE     KIT
010464  032737  004000  177570          BIT     #SW11,@#SWR     ;KILL ITERATIONS
010472  001012                          BNE     SAVLAD
010474  105767  170301                  TSTB    ICNT+1
010500  001404                          BEQ     2$              ;BRANCH IF FIRST
010502  126767  001106  170271          CMPB    TIMES,ICNT+1    ;DONE?
010510  001013                          BNE     KIT             ;BRANCH IF NOT
010512  112767  000001  170261  2$:     MOVB    #1,ICNT+1       ;FIRST ITERATION
010520  105267  170254          SAVLAD: INCB    ICNT            ;COUNT TEST NUMBERS
010524  011667  001060                  MOV     (6),LAD         ;SAVE LOOP ADDRESS
010530  016737  170244  177570          MOV     ICNT,@#DISPLAY  ;DISPLAY TEST NO. AND ITERATION COUNT
010536  000002                          RTI                     ;RETURN

010540  105267  170235          KIT:    INCB    ICNT+1
010544  016737  170230  177570  OVER:   MOV     ICNT,@#DISPLAY  ;SET UP DISPLAY
010552  005767  001032                  TST     LAD             ;FIRST ONE?
010556  001760                          BEQ     SAVLAD
010560  016716  001024                  MOV     LAD,(6)         ;FUDGE RETURN ADDRESS
010564  000002                          RTI                     ;FIXES PS
```

```
010566  032737  002000  177570  .TRP:   BIT     #SW10,2#SWR     ;BELL ON ERROR?
010574  001405                          BEQ     1$              ;NO - SKIP
010576  012767  000207  001000          MOV     #BELL,.TYPE     ;TYPE A BELL
010604  000004  011604                  TYPE    .TYPE
010610  004767  000406          1$:     JSR     PC,ERROR        ;COUNT THE NUMBER OF ERRORS
010614  010446                          MOV     R4,-(6)
010616  032737  020000  177570          BIT     #SW13,2#SWR     ;SKIP TYPEOUT IF SET
010624  001072                          BNE     4$
010626  000004  011552                  TYPE    ,RETURN
010632  016646  000002                  MOV     2(6),-(6)       ;PUT ADDRESS OF INSTRUCTION ON STACK
010636  162716  000002                  SUB     #2,(6)
010642  011605                          MOV     (6),TTY         ;TYPE (6) IN OCTAL
010644  004767  000212                  JSR     %7,PRINTR       ;TYPE LEADING ZERO'S
010650  000004  011560                  TYPE    ,SPACE+3
010654  010005                          MOV     R0,TTY          ;TYPE R0 IN OCTAL
010656  004767  000200                  JSR     %7,PRINTR       ;TYPE LEADING ZERO'S
010662  000004  011561                  TYPE    ,SPACE+4
010666  012703  001002                  MOV     #ANS1,R3        ;ADDRESS OF DATA
010672  113604                          MOVB    @(6)+,R4        ;AMOUNT OF DATA IN TABLE
010674  001426                          BEQ     3$
010676  100016                          BPL     2$              ;TYPE STACK?
010700  016667  000006  170074          MOV     6(6),ANS1
010706  016667  000010  170070          MOV     10(6),ANS2
010714  016667  000012  170064          MOV     12(6),ANS3
010722  016667  000014  170060          MOV     14(6),ANS4
010730  042704  177600                  BIC     #177600,R4      ;CLEAR SIGN
010734  000004  011561          2$:     TYPE    ,SPACE+4
010740  012305                          MOV     (3)+,TTY        ;TYPE (3)+ IN OCTAL
010742  004767  000114                  JSR     %7,PRINTR       ;TYPE LEADING ZERO'S
010746  005304                          DEC     R4
010750  001371                          BNE     2$
010752  005700                  3$:     TST     FPS
010754  100016                          BPL     4$
010756  000004  011555                  TYPE    ,SPACE
010762  170367  170034                  STST    FEC
010766  016705  170030                  MOV     FEC,TTY         ;TYPE FEC IN OCTAL
010772  004767  000064                  JSR     %7,PRINTR       ;TYPE LEADING ZERO'S
010776  000004  011560                  TYPE    ,SPACE+3
011002  016705  170016                  MOV     FEA,TTY         ;TYPE FEA IN OCTAL
011006  004767  000050                  JSR     %7,PRINTR       ;TYPE LEADING ZERO'S
011012  012604                  4$:     MOV     (6)+,R4
011014  005737  177570                  TST     @#SWR           ;HALT ON ERROR
011020  100001                          BPL     .+4             ;SKIP IF CONTINUE
011022  000000                          HALT                    ;HALT ON ERROR!
011024  032737  001000  177570          BIT     #SW09,@#SWR     ;CHECK FOR INHIBIT LOOP ON ERROR
011032  001001                          BNE     .+4             ;SKIP IF LOOP ON ERROR
011034  000002                          RTI
011036  105067  167737                  CLRB    ICNT+1
011042  032737  000400  177570          BIT     #SW08,@#SWR     ;CHECK FOR LOAD MICROBREAK
011050  001233                          BNE     KIT             ;BRANCH IF NOT
011052  113703  177570                  MOVB    @#SWR,R3        ;PUT MICROBREAK ADDRESS IN R3
011056  170003                          LDUB                    ;LOAD MICROBREAK
011060  000627                          BP      KIT             ;LOOP ON TEST UNTIL NO ERRORS
```

```
011062  112767  000001  000130  PRINTR: MOVB    #1,A4$          ;SET ZERO FILL SWITCH
011070  000402                  BR      .+6
011072  005067  000122          PRINTS: CLR     A4$             ;SUPRESS LEADING ZERO'S
011076  112767  177772  000115          MOVB    #-6.A4$+1       ;SET COUNT
011104  010446                  MOV     R4,-(6)         ;SAVE R4
011106  012704  011210                  MOV     #3$,R4          ;SET POINTER TO FIRST ASCII CHAR.
011112  105014                  CLRB    (4)             ;CLEAR FIRST BYTE
011114  000405                  BR      2$              ;ROTATE FIRST BIT
011116  105014          1$:     CLRB    (4)             ;CLEAR BYTE OF CHARACTER
011120  006105                  ROL     TTY             ;ROTATE BIT INTO C
011122  106114                  ROLB    (4)             ;PACK IT
011124  006105                  ROL     TTY             ;ROTATE BIT INTO C
011126  106114                  ROLB    (4)             ;PACK IT
011130  006105          2$:     ROL     TTY             ;ROTATE BIT INTO C
011132  106114                  ROLB    (4)             ;PACK IT
011134  105714                  TSTB    (4)
011136  001402                  BEQ     .+6
011140  105267  000054          INCB    A4$
011144  105767  000050          TSTB    A4$             ;CHECK FILL SWITCH
011150  001402                  BEQ     .+6
011152  152724  000060          BISB    #'0,(4)+        ;MAKE INTO ASCII CHAR
011156  105267  000037          INCB    A4$+1
011162  001355                  BNE     1$              ;REPEAT
011164  022704  011210          CMP     #3$,R4
011170  001002                  BNE     .+6
011172  112724  000060          MOVB    #'0,(4)+
011176  105014                  CLRB    (4)
011200  000004  011210          TYPE    ,3$             ;TYPE IT
011204  012604                  MOV     (6)+,R4         ;RESTORE R4
011206  000207                  RTS     PC

011210  000004          3$:     .BLKW   4
011220  000000          A4$:    0

011222  005267  000364  ERROR:  INC     ERRORS          ;COUNT ERRORS
011226  132737  000001  000041  BITB    #1,@#41         ;AUTO MODE?
011234  001412                  BEQ     1$              ;NO!
011236  022767  000010  000346  CMP     #10.,ERRORS     ;TOO MANY?
011244  001006                  BNE     1$              ;NOT YET
011246  013700  000042          MOV     @#42,R0         ;GET ADDRESS
011252  001403                  BEQ     1$              ;FORGET IT IF ZERO
011254  005037  000042          CLR     @#42            ;ZAP 42
011260  004710                  JSR     PC,(7)          ;CALL THE MONITOR
011262  000207          1$:     RTS     PC              ;RETURN
```

```
011264  012777  011460  000306  POWDWN: MOV     #ILLUP,@UPVEC   ;SET FOR FAST UP
011272  012777  000340  000302          MOV     #340,@UPVEC+2   ;PRIO:7
011300  170246                          STFPS   -(6)            ;GET THE FPS
011302  170011                          SETD
011304  174046                          STD     AC0,-(6)        ;SAVE AC'S
011306  174146                          STD     AC1,-(6)
011310  174246                          STD     AC2,-(6)
011312  174346                          STD     AC3,-(6)
011314  172404                          LDD     AC4,AC0
011316  174046                          STD     AC0,-(6)
011320  172405                          LDD     AC5,AC0
011322  174046                          STD     AC0,-(6)
011324  010046                          MOV     R0,-(6)         ;SAVE REGISTERS
011326  010146                          MOV     R1,-(6)
011330  010246                          MOV     R2,-(6)
011332  010346                          MOV     R3,-(6)
011334  010446                          MOV     R4,-(6)
011336  010546                          MOV     R5,-(6)
011340  010667  000220                  MOV     SP,SAVE6        ;SAVE SP
011344  012777  011354  000226          MOV     #POWUP,@UPVEC   ;SET UP VECTOR
011352  000000                          HALT

011354  016706  000204          POWUP:  MOV     SAVE6,SP        ;GET SP
011360  005001                          CLR     R1              ;WAIT LOOP FOR THE TTY
011362  005201                  1$:     INC     R1
011364  001376                          BNE     1$
011366  012605                          MOV     (6)+,R5         ;GET THE REGISTERS
011370  012604                          MOV     (6)+,R4
011372  012603                          MOV     (6)+,R3
011374  012602                          MOV     (6)+,R2
011376  012601                          MOV     (6)+,R1
011400  012600                          MOV     (6)+,R0
011402  170011                          SETD
011404  172426                          LDD     (6)+,AC0        ;RESTORE THE AC'S
011406  174005                          STD     AC0,AC5
011410  172426                          LDD     (6)+,AC0
011412  174004                          STD     AC0,AC4
011414  172726                          LDD     (6)+,AC3
011416  172626                          LDD     (6)+,AC2
011420  172526                          LDD     (6)+,AC1
011422  172426                          LDD     (6)+,AC0
011424  170126                          LDFPS   (6)+            ;RESTORE FPS
011426  012777  011264  000140          MOV     #POWDWN,@DWNVEC ;SET UP THE POWER DOWN VECTOR
011434  012777  000340  000134          MOV     #340,@DWNVEC+2
011442  000004  011446                  TYPE    ..+2            ;.ASCIZ <15><12>"POWER"
011456  000002                          RTI

011460  000000          ILLUP:  HALT                    ;THE POWER UP SEQUENCE WAS STARTED
011462  000776                  BR      .-2             ; BEFORE THE POWER DOWN WAS COMPLETE
```

```
011464  010546                  .IOT:   MOV     TTY,-(6)        ;SAVE TTY
011466  017605  000002                  MOV     @2(6),TTY       ;GET ADDRESS TO BE TYPED
011472  105715                  1$:     TSTB    (TTY)           ;TERMINATOR?
011474  001406                          BEQ     2$
011476  112537  177566                  MOVB    (TTY)+,@#177566 ;LOAD AND TYPE THE CHARACTER
011502  105737  177564                  TSTB    @#177564        ;IS THE PRINTER READY
011506  100375                          BPL     .-4
011510  000770                          BR      1$              ;GET THE NEXT CHARACTER
011512  017646  000002          2$:     MOV     @2(6),-(6)      ;GET ADDRESS TO BE TYPED
011516  062766  000002  000004          ADD     #2,4(6)         ;ADD 2 TO THE ADDRESS
011524  022666  000002                  CMP     (6)+,2(6)       ;IS IT .+2?
011530  001006                          BNE     3$              ;NO
011532  062705  000002                  ADD     #2,TTY          ;ADD 2 TO THE ADDRESS
011536  042705  000001                  BIC     #1,TTY          ;BACK UP TO AN EVEN BYTE
011542  010566  000002                  MOV     TTY,2(6)        ;RESTORE ADDRESS
011546  012605                  3$:     MOV     (6)+,TTY        ;RESTORE TTY
011550  000002                          RTI                     ;RETURN

011552  005015          000     RETURN: .ASCIZ  <15><12>        ;RETURN AND LINEFEED
011555      015 020012  020040  SPACE:  .ASCIZ  <15><12>"   "   ;RETURN AND 3 SPACES
011562      000
        011564                  .EVEN
011564  000000                  SAVE6:  0
011566  172160                  FPTADR: 172160                  ;FLOATING POINT ADDRESS ON THE 11/20
011570  000244  000246          FPVECT: 244,246                 ;FLOATING POINT VECTOR ADDRESS
011574  000024  000026          DWNVEC: 24,26                   ;POWER DOWN VECTOR ADDRESS
011600  000024  000026          UPVEC:  24,26                   ;POWER UP VECTOR ADDRESS
011604  000000                  .TYPE:  0
011606  000000                  TRPB:   0
011610  000000                  LAD:    0                       ;LOOP ADDRESS
011612  000000                  ERRORS: 0                       ;ERROR COUNT
011614  000377                  TIMES:  377                     ;ITERATION COUNT
        000001                  .END
```

```
AC0     =%000000  391#  584*  585*  591   612*  613*  619   669*  670*  685   735*  736*  751
                  810*  811*  817   866*  868*  874   924*  925   982*  983   1011* 1012  1069*
                  1070  1216*  1218  1289*  1290*  1296  1437*  1438*  1444  1596*  1597*  1612  1725*
                  1727*  1733  1763*  1765*  1771  2079*  2080  2178*  2180  2353  2357*  2358  2359*
                  2360  2382*  2383  2384*  2385  2389*
AC1     =%000001  392#  500*  501*  507   1127*  1128  1252*  1254  1400*  1401*  1407  1549*  1550*
                  1565  1839*  1840  1959*  1960  2019*  2020  2110*  2111  2354  2388*
AC2     =%000002  393#  472*  473*  479   528*  529*  535   556*  557*  563   640*  641*  647
                  706*  707*  713   773*  774*  789   838*  839*  845   895*  897*  903   953*
                  954   1040*  1041  1186*  1187  1363*  1364*  1370  1511*  1512*  1518  1689*  1690*
                  1696  1801*  1803*  1809  1899*  1900  2141*  2143  2355  2387*
AC3     =%000003  394#  1098*  1099  1156*  1157  1326*  1327*  1333  1474*  1475*  1481  1643*  1644*
                  1659  1869*  1870  1929*  1930  1989*  1990  2049*  2050  2356  2386*
AC4     =%000004  395#  2357  2385*
AC5     =%000005  396#  2359  2383*
ANR32   004170  1217*  1219#  1231
ANR33   004310  1253*  1255#  1267
ANR64   010104  2142*  2144#  2156
ANR65   010230  2179*  2181#  2193
ANS1    001002  425#  479*  480   507*  508   535*  536   563*  564   591*  592   619*  620
                  647*  648   685*  686   713*  714   751*  752   789*  790   917*  818   845*
                  846   874*  875   903*  904   925*  933   954*  962   983*  991   1012*  1020
                  1041*  1049  1070*  1078  1099*  1107  1128*  1136  1160*  1166  1190*  1196  1231*
                  1232  1267*  1268  1296*  1297  1333*  1334  1370*  1371  1407*  1408  1444*  1445
                  1481*  1482  1518*  1519  1565*  1566  1612*  1613  1659*  1660  1696*  1697  1733*
                  1734  1771*  1772  1809*  1810  1840*  1848  1870*  1878  1900*  1908  1930*  1939
                  1960*  1968  1990*  1998  2020*  2028  2050*  2058  2083*  2089  2114*  2120  2156*
                  2157  2193*  2194  2269  2273*
ANS2    001004  426#  484   512   540   568   596   624   652   690   718   756   794   822
                  850   879   908   926*  927*  937   955*  956*  966   984*  985*  995   1013*
                  1014*  1024  1042*  1043*  1053  1071*  1072*  1082  1100*  1101*  1111  1129*  1130*
                  1140  1158*  1159*  1170  1188*  1189*  1200  1224*  1225*  1236  1260*  1261*  1272
                  1301  1338  1375  1412  1449  1486  1523  1570  1617  1664  1701  1738  1776
                  1814  1841*  1842*  1852  1871*  1872*  1882  1901*  1902*  1912  1931*  1932*  1942
                  1961*  1962*  1972  1991*  1992*  2002  2021*  2022*  2032  2051*  2052*  2062  2081*
                  2082*  2093  2112*  2113*  2124  2149*  2150*  2161  2186*  2187*  2198  2274*
ANS3    001006  427#  1305  1342  1379  1416  1453  1490  1527  1574  1621  1668  1705  1742
                  1780  1818  2275*
ANS4    001010  428#  1309  1346  1383  1420  1457  1494  1531  1578  1625  1672  1709  1746
                  1784  1822  2276*
ANS5    001012  429#
ANS6    001014  430#
ANS7    001016  431#
ANS8    001020  432#
R4$     011220  2305*  2307*  2308*  2322*  2323  2326*  2337#
BEG     001026  415   436#
BEGIN   001062  440   445#  2214  2216
BELL    =000207  450#  2206  2254
DAT1    001200  469#  472
DAT10   001772  666#  669
DAT11   002110  703#  706
DAT12   002176  732#  735
DAT13   002314  770#  773
DAT14   002432  807#  810
DAT15   002520  835#  838
DAT16   002606  863#  866
```

```
DAT17   002676          892#    895
DAT2    001266          497#    500
DAT20   002766          921#    924
DAT21   003064          950#    953
DAT22   003162          979#    982
DAT23   003260         1008#   1011
DAT24   003356         1037#   1040
DAT25   003454         1066#   1069
DAT26   003552         1095#   1098
DAT27   003650         1124#   1127
DAT3    001354          525#    528
DAT30   003746         1153#   1156
DAT31   004046         1183#   1186
DAT32   004146         1213#   1216
DAT33   004266         1249#   1252
DAT34   004406         1285#   1289
DAT35   004524         1322#   1326
DAT36   004642         1359#   1363
DAT37   004760         1396#   1400
DAT4    001442          553#    556
DAT40   005076         1433#   1437
DAT41   005214         1470#   1474
DAT42   005332         1507#   1511
DAT43   005460         1545#   1549
DAT44   005616         1592#   1596
DAT45   005764         1639#   1643
DAT46   006132         1685#   1689
DAT47   006250         1722#   1726
DAT5    001530          581#    584
DAT50   006366         1759#   1763
DAT51   006506         1797#   1801
DAT52   006626         1835#   1839
DAT53   006730         1865#   1869
DAT54   007032         1895#   1899
DAT55   007134         1925#   1929
DAT56   007236         1955#   1959
DAT57   007340         1985#   1989
DAT6    001616          609#    612
DAT60   007442         2015#   2019
DAT61   007544         2045#   2049
DAT62   007646         2075#   2079
DAT63   007752         2106#   2110
DAT64   010056         2137#   2141
DAT65   010202         2174#   2178
DAT7    001704          637#    640
DISPLA= 177570          376#   2243*   2247*
DONE    010322         2203#
DWNVEC  011574          448*    449*   2391*   2392*   2423#
ERROR   011222         2256    2339#
ERRORS  011612         2339*   2342    2428#
FEA     001024          434#    681     747     785    1561    1608    1655    2290
FEC     001022          419*    433#    672*    677     738*    743     776*    781    1552*   1557    1599*   1604    1646*
                       1651    2286*   2287
F_TERR  000760          418#    456
FPI1    001214          473#
FPI10   002006          670#    681
```

DCFPKB.P11 CROSS REFERENCE TABLE -- USER SYMBOLS

```
FPI11   002124          707#
FPI12   002212          736#    747
FPI13   002330          774#    785
FPI14   002446          811#
FPI15   002534          839#
FPI16   002626          868#
FPI17   002716          897#
FPI2    001302          501#
FPI20   003002          925#
FPI21   003100          954#
FPI22   003176          983#
FPI23   003274         1012#
FPI24   003372         1041#
FPI25   003470         1070#
FPI26   003566         1099#
FPI27   003664         1128#
FPI3    001370          529#
FPI32   004166         1218#
FPI33   004306         1254#
FPI34   004426         1290#
FPI35   004544         1327#
FPI36   004662         1364#
FPI37   005000         1401#
FPI4    001456          557#
FPI40   005116         1438#
FPI41   005234         1475#
FPI42   005352         1512#
FPI43   005470         1550#   1561
FPI44   005636         1597#   1608
FPI45   006004         1644#   1655
FPI46   006152         1690#
FPI47   006270         1727#
FPI5    001544          585#
FPI50   006412         1765#
FPI51   006532         1803#
FPI6    001632          613#
FPI7    001720          641#
FPS    =%000000         381#    418*    474*    475     502*    503     530*    531     558*    559     586*    587     614*
                        615     642*    643     671*    673     708*    709     737*    739     775*    777     812*    813
                        840*    841     869*    870     898*    899     928*    929     957*    958     986*    987     1015*
                       1016    1044*   1045    1073*   1074    1102*   1103    1131*   1132    1161*   1162    1191*   1192
                       1226*   1227    1262*   1263    1291*   1292    1328*   1329    1365*   1366    1402*   1403    1439*
                       1440    1476*   1477    1513*   1514    1551*   1553    1598*   1600    1645*   1647    1691*   1692
                       1728*   1729    1766*   1767    1804*   1805    1843*   1844    1873*   1874    1903*   1904    1933*
                       1934    1963*   1964    1993*   1994    2023*   2024    2053*   2054    2084*   2085    2115*   2116
                       2151*   2152    2188*   2189    2283

FPTADR  011566          442     2421#
FPVECT  011570          456*    457*    2422#
HLT    = 104000         378#    477     482     486     505     510     514     533     538     542     561     566     570
                        589     594     598     617     622     626     645     650     654     675     679     683     688
                        692     711     716     720     741     745     749     754     758     779     783     787     792
                        796     815     820     824     843     848     852     872     877     881     901     906     910
                        931     935     939     960     964     968     989     993     997    1018    1022    1026    1047
                       1051    1055    1076    1080    1084    1105    1109    1113    1134    1138    1142    1164    1169
                       1172    1194    1198    1202    1229    1234    1238    1265    1270    1274    1294    1299    1303
                       1307    1311    1331    1336    1340    1344    1348    1368    1373    1377    1381    1385    1405
```

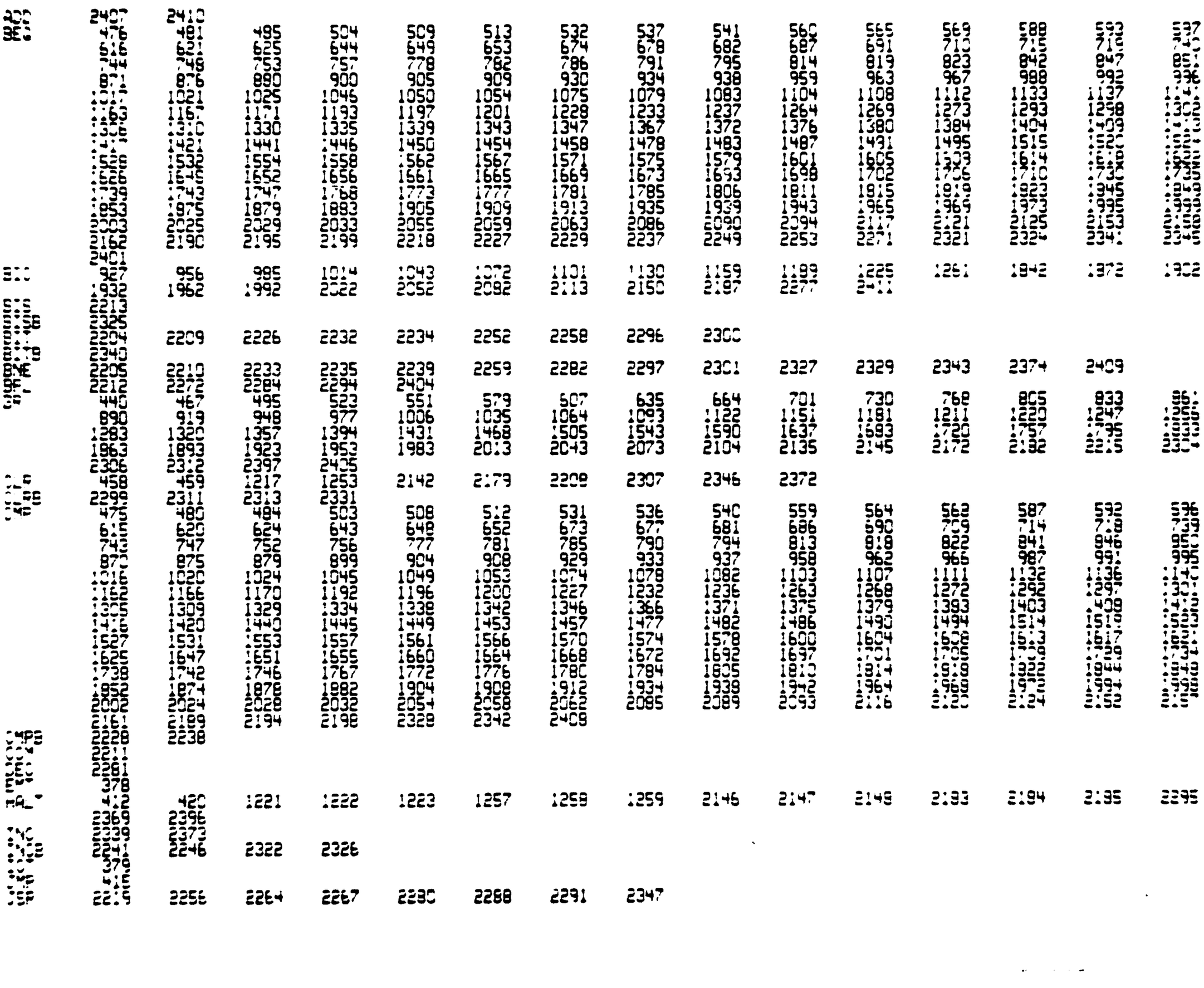
```
                    1410    1414    1418    1422    1442    1447    1451    1455    1459    1479    1484    1488    1492
                    1496    1516    1521    1525    1529    1533    1555    1559    1563    1568    1572    1576    1580
                    1602    1606    1610    1615    1619    1623    1627    1649    1653    1657    1662    1666    1670
                    1674    1694    1699    1703    1707    1711    1731    1736    1740    1744    1748    1769    1774
                    1778    1782    1786    1807    1812    1816    1820    1824    1846    1850    1854    1876    1880
                    1884    1906    1910    1914    1936    1940    1944    1966    1970    1974    1996    2000    2004
                    2026    2030    2034    2056    2060    2064    2087    2091    2095    2118    2122    2126    2154
                    2159    2163    2191    2196    2200
ICNT    001000       424*    458*    2228    2236    2238    2240*   2241*   2243    2246*   2247    2299*
ILLUP   011460      2349    2396*
KBT     010540      2233    2239    2246*   2301    2304
LAD     011610       459*   2242*   2248    2250    2427*
LDSC  = 170004       409*
LDJB  = 170003       405*
MRS   = 170006       408*
MTI20   001054       437     442*
N     = 000066       373*    460     488*    516*    544*    572*    600*    628*    656*    694*    722*    760*    798*
                     826*    854*    883*    912*    941*    970*    999*   1028*   1057*   1086*   1115*   1144*   1174*
                    1204*   1240*   1276*   1313*   1350*   1387*   1424*   1461*   1498*   1535*   1592*   1629*   1676*
                    1713*   1750*   1788*   1826*   1856*   1886*   1916*   1946*   1976*   2006*   2036*   2066*   2097*
                    2128*   2165*   2202*
OVER    010544      2229    2247*
PC    =%000007       390*   1218*   1254*   2143*   2180*   2256*   2334*   2347*   2348*
POWDWN  011264       448    2349*   2391
POWUP   011354      2368    2371*
PRINTR  011062      2264    2267    2280    2288    2291    2305*
PRINTS  011072      2307*
PS    = 177776       374*    926     955     984    1013    1042    1071    1100    1129    1158    1188    1224    1260
                    1841    1871    1901    1931    1961    1991    2021    2051    2081    2112    2149    2186
RETURN  011552      2260    2416*
R0    =%000000       382*    451*   2217*   2266    2344*   2361    2380*
R1    =%000001       383*   1157*   1160    2362    2372*   2373*   2379*
R2    =%000002       384*    867*    868    1802*   1803    2363    2378*
R3    =%000003       385*    896*    897    2080*   2083    2230*   2269*   2302*   2364    2377*
R4    =%000004       386*   1187*   1190    2111*   2114    2257    2270*   2277*   2281*   2292*   2309    2310*   2329
                    2333*   2365    2376*
R5    =%000005       387*   1764*   1765    2366    2375*
SAVE6   011564      2367*   2371    2420*
SAVLAD  010520      2235    2241*   2249
SCOPE = 104400       377*    466     494     522     550     578     606     634     663     700     729     767     804
                     832     860     889     918     947     976    1005    1034    1063    1092    1121    1150    1180
                    1210    1246    1282    1319    1356    1393    1430    1467    1504    1542    1589    1636    1682
                    1719    1756    1794    1832    1862    1892    1922    1952    1982    2012    2042    2072    2103
                    2134    2171    2203
SP    =%000006       389*    436*    446*   2367    2371*
SPACE   011555      2265    2268    2278    2285    2289    2417*
STA0  = 170005       406*
STQ0  = 170007       407*
SWR   = 177570       375*    376     2204    2209    2226    2228    2230    2232    2234    2252    2258    2293    2296
                    2300    2302
SW08  = 000400       404*   2226    2300
SW09  = 001000       403*   2296
SW10  = 002000       402*   2204    2252
SW11  = 004000       401*   2234
SW12  = 010000       400*   2209
SW13  = 020000       399*   2258
```

DCFPKB.P11 CROSS REFERENCE TABLE -- USER SYMBOLS

```
SW14  = 040000            398*   2232
SW15  = 100000            397*
TIMES   011614           2238    2429*
TRPB    011606           2211*   2426*
TST1    001204            467     471*
TST10   001776            664     668*
TST11   002114            701     705*
TST12   002202            730     734*
TST13   002320            768     772*
TST14   002436            805     809*
TST15   002524            833     837*
TST16   002612            861     865*
TST17   002702            890     894*
TST2    001272            495     499*
TST20   002772            919     923*
TST21   003070            948     952*
TST22   003166            977     981*
TST23   003264           1006    1010*
TST24   003362           1035    1039*
TST25   003460           1064    1068*
TST26   003556           1093    1097*
TST27   003654           1122    1126*
TST3    001360            523     527*
TST30   003752           1151    1155*
TST31   004052           1181    1185*
TST32   004152           1211    1215*
TST33   004272           1247    1251*
TST34   004416           1283    1288*
TST35   004534           1320    1325*
TST36   004652           1357    1362*
TST37   004770           1394    1399*
TST4    001446            551     555*
TST40   005106           1431    1436*
TST41   005224           1468    1473*
TST42   005342           1505    1510*
TST43   005460           1543    1548*
TST44   005626           1590    1595*
TST45   005774           1637    1642*
TST46   006142           1683    1688*
TST47   006260           1720    1725*
TST5    001534            579     583*
TST50   006376           1757    1762*
TST51   006516           1795    1800*
TST52   006636           1833    1838*
TST53   006740           1863    1868*
TST54   007042           1893    1898*
TST55   007144           1923    1928*
TST56   007245           1953    1958*
TST57   007350           1983    1988*
TST6    001622            607     611*
TST60   007452           2013    2018*
TST61   007554           2043    2048*
TST62   007656           2073    2078*
TST63   007762           2104    2109*
TST64   010066           2135    2140*
TST65   010212           2172    2177*
```

BOS

	372#	2263	2266	2279	2287	2290									
	373#	2393													
	374#														
	375#														
	376#														
	377#														
US	400#	474	502	530	558	586	614	642	671	708	737	775	812	840	863
	401#	898	926	954	986	1015	1044	1073	1102	1131	1161	1191	1226	1262	1291	1320
	402#	1365	1402	1439	1476	1513	1551	1598	1645	1691	1728	1766	1804	1843	1873	1923
	403#	1933	1963	1993	2023	2053	2084	2115	2151	2188						
	404#		2206	2254												
	405#		1276	1313	1350	1387	1424	1461	1498	1535	1582	1629	1676	1713		
	406#		488	516	544	572	600	628	656	694	722	750	788	826		
	407#		1760	1788												
	410#		864	884												
	411#		1826	1856	1886	1916	1946	1976	2006	2036						
	412#		912	941	970	999	1028	1057	1086	1115						
	413#		2066	2094												
	414#		2126	2165												
	415#		1144	1174												
	416#		1204	1243												



DOFPAB.P11 CROSS REFERENCE TABLE -- PERMANENT SYMBOLS

```
LDD      1299  1326  1363  1400  1437  1474  1511  1549  1596  1643  1689  1726  1763  1801  1839
         1869  1899  1929  1959  1989  2019  2049  2079  2110  2141  2178  2357  2359  2382  2384
         2386  2387  2388  2389
LDEXP     473   501   529   557   585   613   641   670   707   736   774   811   839   868   897
         1290  1327  1364  1401  1438  1475  1512  1550  1597  1644  1690  1727  1765  1803
LDF       472   500   528   556   584   612   640   669   706   735   773   810   838   866   895
          924   953   982  1011  1040  1069  1098  1127  1156  1186  1216  1252
LDFPS     471   499   527   555   583   611   639   668   705   734   772   809   837   865   894
          923   952   981  1010  1039  1068  1097  1126  1155  1185  1215  1251  1288  1325  1363
         1399  1436  1473  1510  1548  1595  1642  1688  1725  1762  1800  1838  1876  1913  1951
         1958  1988  2018  2048  2078  2109  2140  2177  2390
LDLB     2231  2303
LDST      436   437   439   442   445   446   447   448   449   450   451   452   453   454   455
          456   457   867   896   926   955   984  1013  1042  1071  1100  1129  1158  1186  1188
         1190  1224  1231  1260  1267  1764  1802  1841  1871  1901  1931  1961  1991  2021  2051
         2081  2083  2112  2114  2149  2156  2186  2193  2206  2214  2216  2217  2242  2243  2247
         2250  2254  2257  2261  2263  2266  2269  2273  2274  2275  2276  2279  2287  2290  2292
         2309  2310  2333  2344  2349  2350  2361  2362  2363  2364  2365  2366  2367  2368  2371
         2375  2376  2377  2378  2379  2380  2391  2392  2398  2399  2406  2412  2413
MOVB     2230  2240  2270  2302  2305  2308  2330  2402
NOP      2220  2221  2222
ROL      2314  2316  2318
ROLB     2315  2317  2319
RTI       421  2223  2224  2244  2251  2298  2394  2414
RTS      2334  2348
SETD     2352  2381
STD      1296  1333  1370  1407  1444  1481  1518  1565  1612  1659  1696  1733  1771  1809  2353
         2354  2355  2356  2358  2360  2383  2385
STEXP     925   954   983  1012  1041  1070  1099  1128  1157  1187  1218  1254  1840  1870  1900
         1930  1960  1990  2020  2050  2080  2111  2143  2180
STF       479   507   535   563   591   619   647   685   713   745   788   817   844   874   903
STFPS     418   474   502   530   558   586   614   642   671   708   737   775   812   842   873
          898   928   957   986  1015  1044  1073  1102  1131  1161  1191  1226  1266  1301  1341
         1365  1402  1439  1476  1513  1551  1598  1645  1691  1728  1766  1804  1843  1881  1920
         1933  1963  1993  2023  2053  2084  2115  2151  2189  2351
STST      419   672   738   776  1552  1599  1646  2286
SUBB     2262
SUBP      377
TST       438  2248  2283  2293
TSTB     2236  2320  2323  2400  2403
.ASCIZ   2394  2416  2417
.BLKW    2336
.ENABL    372
.END     2430
.ENDC     464   475   479   492   503   507   520   531   535   548   559   563   576   587   631
          604   615   619   632   643   647   661   673   685   698   709   733   746   759   803
          765   777   789   802   813   817   830   841   845   858   870   874   887   899   903
          929   933   958   962   987   991  1016  1020  1045  1049  1074  1078  1103  1107  1133
         1136  1162  1166  1192  1196  1227  1231  1263  1267  1280  1292  1296  1317  1329  1333
         1364  1366  1370  1391  1403  1407  1428  1440  1444  1465  1477  1481  1502  1514  1519
         1540  1553  1565  1587  1600  1612  1634  1647  1659  1680  1692  1696  1717  1729  1733
         1754  1767  1771  1792  1805  1809  1844  1848  1874  1878  1904  1908  1934  1938  1964
         1988  1994  1998  2024  2028  2054  2058  2085  2089  2116  2120  2152  2156  2189  2193
.EOT     2338  2419
.PSECT    464   475   492   503   520   531   548   559   576   587   604   615   632   643   661
          672   698   725   738   764   776   802   813   830   841   858   869   886   897   914
```

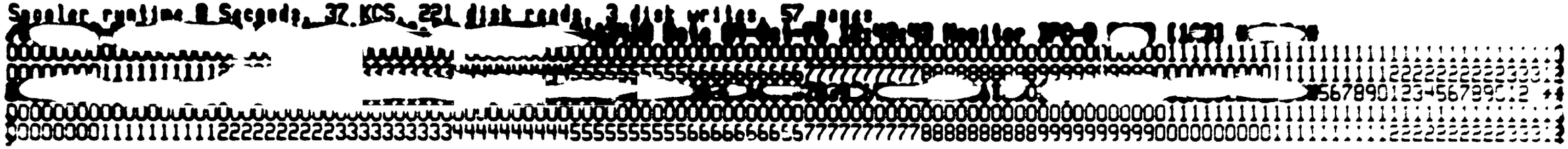
F05


```
                929     958     987    1016    1045    1074    1103    1132    1162    1192    1227    1263    1280    1292    1317
               1329    1354    1366    1391    1403    1428    1440    1465    1477    1502    1514    1539    1552    1586    1599
               1633    1646    1680    1692    1717    1729    1754    1767    1792    1805    1844    1874    1904    1934    1964
               1994    2024    2054    2085    2116    2152    2189
.DPN            479     507     535     563     591     619     647     677     681     713     743     747     781     785     817
                945     874     903     933     962     991    1020    1049    1078    1107    1136    1166    1196    1231    1267
               1296    1333    1370    1407    1444    1481    1518    1557    1561    1604    1608    1651    1655    1696    1733
               1772    1809    1848    1879    1908    1938    1968    1998    2028    2058    2089    2120    2156    2193
.DPSY           334     372     412     422     460     488     516     544     572     600     628     656     694     722     760
               1798     826     854     882     912     941     970     999    1028    1057    1086    1115    1144    1174    1204
               1240    1276    1313    1350    1387    1424    1461    1498    1535    1582    1629    1676    1713    1750    1788
               1826    1856    1886    1916    1946    1976    2006    2036    2066    2097    2128    2165    2202    2252    2305
               2349    2394    2398
.DPSN           372     460
.DPSNS          376
.DPSNU          327     372     412     422     460     488     516     544     572     600     628     656     694     722     760
               1798     826     854     882     912     941     970     999    1028    1057    1086    1115    1144    1174    1204
               1240    1276    1313    1350    1387    1424    1461    1498    1535    1582    1629    1676    1713    1750    1788
               1826    1856    1886    1916    1946    1976    2006    2036    2066    2097    2128    2165    2202    2252    2305
               2349    2394    2398
.DPSS           460    2202
.DPSSE          331
.DPSSF           32     412
.DPSSU          327     372     422     460    2202    2252    2305    2349    2398
.DPSUE          328
```

```
ERRORS DETECTED:  0
DEFAULT GLOBALS GENERATED:  0

*DCFPKB,DCFPKB.SEQ/SOL/CRF/DS:ERFZ/EN:ABS=DSKM:DCFPKB.P11
RUN-TIME:  16  24  3 SECONDS
RUN-TIME RATIO:  2797/46=60.8
CORE USED:  10K  (20 PAGES)
```

GO5


Spooler runtime 8 Seconds, 37 KCS, 221 disk reads, 3 disk writes, 57 pages