

FP11

LDCDF LDCFP STCFD STCDF
MD-11-DCFPI-B

EP DCFPI B DL A

OCT 1976

COPYRIGHT © 1976

digital

FICHE 1 OF 1

Made in U.S.A.

The microfiche card displays a grid of 48 frames, organized into 8 rows and 6 columns. Each frame contains a small, high-contrast image, likely a technical drawing or data table, which is difficult to read due to the small size and high contrast. The frames are arranged in a regular grid pattern on the left side of the card.

.REPT 0

IDENTIFICATION

PRODUCT CODE: MAINDEC-11-DCFP1
 PRODUCT NAME: FP11 BASIC INSTRUCTION TESTS
 DATE CREATED: MARCH 12, 1973
 MAINTAINER: DIAGNOSTIC GROUP
 AUTHORS: BOB BRAIN & KEN CHAPMAN

COPYRIGHT (C) DIGITAL EQUIPMENT CORPORATION
 1973

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

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

MAINDEC-11-DCFP1-B
 11-DCFP1-B

E01

MAINDEC-11-DOFPI-B
DOFPI.P11

TEST OF LDCDF, LDCFD, STCFD, STCDF

MACY11 27(732) 17-SEP-76 10:46 PAGE 4

168
168

7) THE DISPLAY ON THE 11/45 WILL SHOW THE ITERATION COUNT IN
THE LEFT BYTE AND TEST NUMBER IN THE RIGHT. TO USE, SET THE

FP11 BASIC INSTRUCTION TEST DCFPA - DCFPL
DESCRIPTION

PAGE 5

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 RD.

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.

MAY 1977
NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY
GALVESTON OFFICE
77058

01
02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

FP11 BASIC INSTRUCTION TEST DCFPA - DCFPL
DESCRIPTION

PAGE 6

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 INITIALLY 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.
.ENDR

.TITLE MAINDEC-11-DCFPI-B TEST OF LDCDF, LDCFD, STCFD, STCDF
: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 ERROR TYPEOUTS
14	LOOP ON TEST
15	HALT ON ERROR

OUTPUT FORM:

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

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

```

000001      .ENABL  ABS
177776      N=      1
177570      PS=     177776
177570      SWR=     177570
104400      DISPLAY=SWR
104000      SCOPE=   TRAP
000004      HLT=     EMT
000207      TYPE=    IOT
000000      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      ACO=     %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      STAO=    170005
170007      STGO=    170007
170006      MRS=    170006
170004      LDSC=    170004

000000      .=      0
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

```

;TRAP CATCHER FROM 0 - 776

```
001000 001000      . =      1000
001000 000000      ICNT:    0
001002 000000      ANS1:    0
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
001024 000000      FEA:      0
                                ; ITERATION COUNT - LH TEST NO. - RH
                                ; FIRST ANSWER (SEE CODE)

                                ; FLOATING EXCEPTION CODES
                                ; FLOATING EXECPTION ADDRESS

001026 012706 000600      BEG:    MOV    #600, SP      ; ** STACK AT 600 **
001032 012737 001054 000004      MOV    #M1120, a#4  ; FIND OUT WHICH MACHINE THIS IS
001040 005737 177772      TST    a#177772    ; IS PIRQ THERE?
001044 012767 000006 015144      MOV    #6, YESRT   ; FUDGE IN RTT IF 11/45
001052 000403      BR     BEGIN

001054 016737 016300 000010      M1120: MOV    FPTADR, a#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, a#4      ; RESET 4
001070 012706 000600      MOV    #600, SP
001074 012737 016216 000014      MOV    #YESRT, a#14 ; SET TRACE TRAP VECTOR
001102 012777 017056 016256      MOV    #POWDWN, aDOWNVEC
001110 012777 000340 016252      MOV    #340, aDOWNVEC+2
001116 012737 017256 000020      MOV    #.IOT, a#20  ; SET UP VECTOR 20
001124 012700 000030      MOV    #30, R0      ; SET R0 TO VECTOR 30
001130 012720 016360      MOV    #.TRP, (0)+  ; SET EMT VECTOR
001134 012720 000340      MOV    #340, (0)+
001140 012720 016220      MOV    #.EMT, (0)+  ; SET TRAP VECTOR
001144 012710 000340      MOV    #340, (0)
001150 012777 000760 016204      MOV    #FLTERR, aFPVECT ; LOAD INTERUPT VECTOR
001156 012777 000340 016200      MOV    #340, aFPVECT+2 ; LOCK UP PROCESSOR
001164 005067 177610      CLR    ICNT
001170 005067 016206      CLR    LAC
```

```

001174 000404 BR TSTO ;SKIP "GARBAGE" DATA
001176 142356 021347 076721 DATO: 142356,021347,076721,136514
001204 136514

```

```

001206 170127 000200 TSTO: LDFPS #000200 ;SET D ONLY
001212 172467 177760 LDD DATO, AC0 ;LOAD "GARBAGE" INTO AC0
001216 174004 STD AC0, AC4 ;STORE THE "GARBAGE" IN AC4

```

```

:*****
:TEST 1: TEST LDCDF (LOAD-CONVERT DOUBLE TO FLOATING)
:LOAD 000000,000000,000000,000000 --> 000000,000000
:FPS = 047404, SRC = M6-R7, AC = AC3
:*****

```

```

001220 104400 SCOPE
001222 000404 BR TST1

001224 000000 000000 000000 DAT1: 000000,000000,000000,000000
001232 000000

```

```

001234 170127 047417 TST1: LDFPS #047417 ;LOAD FLOATING POINT STAUS
001240 177767 177760 FPI1: LDCDF DAT1, AC3 ;LOAD-CONVERT 000000,000000,000000,000000
001244 170200 STFPS FPS ;STORE FLOATING POINT STATUS
001246 022700 047404 CMP #047404,FPS ;CHECK FLOATING POINT STATUS
001252 001401 BEQ .+4 ;BRANCH IF OK
001254 104000 HLT ;FPS NOT EQUAL TO 047404

```

```

001256 174367 177520 STF AC3, ANS1 ;STORE AC3 IN ANS1, ANS2
001262 022767 000000 177512 CMP #000000,ANS1 ;DID 000000 GET STORED?
001270 001401 BEQ .+4 ;BRANCH IF OK
001272 104002 HLT+2 ;ANS1 NOT EQUAL TO 000000

```

```

001274 022767 000000 177502 CMP #000000,ANS2 ;DID 000000 GET STORED?
001302 001401 BEQ .+4 ;BRANCH IF OK
001304 104002 HLT+2 ;ANS2 NOT EQUAL TO 000000

```

```

:*****
:TEST 2: TEST LDCDF (LOAD-CONVERT DOUBLE TO FLOATING)
:LOAD 052525,052525,052525,052525 --> 052525,052525
:FPS = 047400, SRC = M6-R7, AC = AC1
:*****

```

```

001306 104400 SCOPE
001310 000404 BR TST2

001312 052525 052525 052525 DAT2: 052525,052525,052525,052525
001320 052525

```

```

001322 170127 047417 TST2: LDFPS #047417 ;LOAD FLOATING POINT STAUS
001326 177567 177760 FPI2: LDCDF DAT2, AC1 ;LOAD-CONVERT 052525,052525,052525,052525
001332 170200 STFPS FPS ;STORE FLOATING POINT STATUS
001334 022700 047400 CMP #047400,FPS ;CHECK FLOATING POINT STATUS
001340 001401 BEQ .+4 ;BRANCH IF OK
001342 104000 HLT ;FPS NOT EQUAL TO 047400

```

```

001344 174167 177432          STF      AC1,      ANS1      ;STORE AC1 IN ANS1, ANS2
001350 022767 052525 177424    CMP      #052525,ANS1    ;DID 052525 GET STORED?
001356 001401          BEQ      .+4            ;BRANCH IF OK
001360 104002          HLT+2          ;ANS1 NOT EQUAL TO 052525

001362 022767 052525 177414    CMP      #052525,ANS2    ;DID 052525 GET STORED?
001370 001401          BEQ      .+4            ;BRANCH IF OK
001372 104002          HLT+2          ;ANS2 NOT EQUAL TO 052525

```

```

:*****
:TEST 3:      TEST LDCDF (LOAD-CONVERT DOUBLE TO FLOATING)
:      LOAD 125252,125252,125252,125252 --> 125252,125253
:      FPS = 047410, SRC = M6-R7, AC = AC3
:*****

```

```

001374 104400          SCOPE
001376 000404          BR      TST3

001400 125252 125252 125252 DAT3: 125252,125252,125252,125252

001410 170127 047417          TST3:  LDFPS  #047417      ;LOAD FLOATING POINT STAUS
001414 177767 177760          FPI3:  LDCDF  DAT3,   AC3      ;LOAD-CONVERT 125252,125252,125252,125252
001420 170200          STFPS  FPS          ;STORE FLOATING POINT STATUS
001422 022700 047410          CMP    #047410,FPS    ;CHECK FLOATING POINT STATUS
001426 001401          BEQ    .+4            ;BRANCH IF OK
001430 104000          HLT                    ;FPS NOT EQUAL TO 047410

001432 174367 177344          STF    AC3,      ANS1      ;STORE AC3 IN ANS1, ANS2
001436 022767 125252 177336    CMP    #125252,ANS1    ;DID 125252 GET STORED?
001444 001401          BEQ    .+4            ;BRANCH IF OK
001446 104002          HLT+2          ;ANS1 NOT EQUAL TO 125252

001450 022767 125253 177326    CMP    #125253,ANS2    ;DID 125253 GET STORED?
001456 001401          BEQ    .+4            ;BRANCH IF OK
001460 104002          HLT+2          ;ANS2 NOT EQUAL TO 125253

```

```

:*****
:TEST 4:      TEST LDCDF (LOAD-CONVERT DOUBLE TO FLOATING)
:      LOAD 000200,000000,000000,000000 --> 000200,000000
:      FPS = 047400, SRC = M6-R7, AC = AC0
:*****

```

```

001462 104400          SCOPE
001464 000404          BR      TST4

001466 000200 000000 000000 DAT4: 000200,000000,000000,000000

001476 170127 047417          TST4:  LDFPS  #047417      ;LOAD FLOATING POINT STAUS
001502 177467 177760          FPI4:  LDCDF  DAT4,   AC0      ;LOAD-CONVERT 000200,000000,000000,000000
001506 170200          STFPS  FPS          ;STORE FLOATING POINT STATUS
001510 022700 047400          CMP    #047400,FPS    ;CHECK FLOATING POINT STATUS
001514 001401          BEQ    .+4            ;BRANCH IF OK

```

```

001516 104000          HLT          ;FPS NOT EQUAL TO 047400
001520 174067 177256   STF          ACO, ANS1      ;STORE ACO IN ANS1, ANS2
001524 022767 000200 177250 CMP          #000200,ANS1 ;DID 000200 GET STORED?
001532 001401          BEQ          .+4        ;BRANCH IF OK
001534 104002          HLT+2        ;ANS1 NOT EQUAL TO 000200
001536 022767 000000 177240 CMP          #000000,ANS2 ;DID 000000 GET STORED?
001544 001401          BEQ          .+4        ;BRANCH IF OK
001546 104002          HLT+2        ;ANS2 NOT EQUAL TO 000000

```

```

:*****
:TEST 5:      TEST LDCDF (LOAD-CONVERT DOUBLE TO FLOATING)
:      LOAD   100200,000000,000000,000000 --> 100200,000000
:      FPS = 047410, SRC = M6-R7, AC = AC2
:*****

```

```

001550 104400          SCOPE
001552 000404          BR          TST5
001554 100200 000000 000000 DAT5: 100200,000000,000000,000000
001562 000000

```

```

001564 170127 047417   TST5:  LDFPS  #047417      ;LOAD FLOATING POINT STATUS
001570 177667 177760   FP15:  LDCDF  DAT5, AC2      ;LOAD-CONVERT 100200,000000,000000,000000
001574 170200          STFPS  FPS              ;STORE FLOATING POINT STATUS
001576 022700 047410   CMP    #047410,FPS      ;CHECK FLOATING POINT STATUS
001602 001401          BEQ    .+4              ;BRANCH IF OK
001604 104000          HLT          ;FPS NOT EQUAL TO 047410

```

```

001606 174267 177170   STF    AC2, ANS1        ;STORE AC2 IN ANS1, ANS2
001612 022767 100200 177162 CMP    #100200,ANS1     ;DID 100200 GET STORED?
001620 001401          BEQ    .+4              ;BRANCH IF OK
001622 104002          HLT+2        ;ANS1 NOT EQUAL TO 100200
001624 022767 000000 177152 CMP    #000000,ANS2     ;DID 000000 GET STORED?
001632 001401          BEQ    .+4              ;BRANCH IF OK
001634 104002          HLT+2        ;ANS2 NOT EQUAL TO 000000

```

```

:*****
:TEST 6:      TEST LDCDF (LOAD-CONVERT DOUBLE TO FLOATING)
:      LOAD   000177,177777,177777,177777 --> 000000,000000
:      FPS = 047404, SRC = M6-R7, AC = ACO
:*****

```

```

001636 104400          SCOPE
001640 000404          BR          TST6
001642 000177 177777 177777 DAT6: 000177,177777,177777,177777
001650 177777

```

```

001652 170127 047417   TST6:  LDFPS  #047417      ;LOAD FLOATING POINT STATUS
001656 177467 177760   FP16:  LDCDF  DAT6, ACO      ;LOAD-CONVERT 000177,177777,177777,177777
001662 170200          STFPS  FPS              ;STORE FLOATING POINT STATUS

```

```

001664 022700 047404      CMP      #047404,FPS      :CHECK FLOATING POINT STATUS
001670 001401      BEQ      .+4            :BRANCH IF OK
001672 104000      HLT                        :FPS NOT EQUAL TO 047404

001674 174067 177102      STF      ACC,ANS1      :STORE ACC IN ANS1,ANS2
001700 022767 000000 177074      CMP      #000000,ANS1   :DID 000000 GET STORED?
001706 001401      BEQ      .+4            :BRANCH IF OK
001710 104002      HLT+2                    :ANS1 NOT EQUAL TO 000000

001712 022767 000000 177064      CMP      #000000,ANS2   :DID 000000 GET STORED?
001720 001401      BEQ      .+4            :BRANCH IF OK
001722 104002      HLT+2                    :ANS2 NOT EQUAL TO 000000

```

```

:*****
:TEST 7: TEST LDCDF (LOAD-CONVERT DOUBLE TO FLOATING)
:LOAD 100177,177777,177777,177777 --> 000000,000000
:FPS = 147414, SRC = M6-R7, AC = ACC
:FEC = 14, FEA = FPI7
:*****

```

```

001724 104400      SCOPE
001726 000406      BR      TST7

001730 000000 000000      SET7: 000000,000000
001734 100177 177777 177777      DAT7: 100177,177777,177777,177777
001742 177777

001744 170127 047417      TST7: LDFPS #047417      :LOAD FLOATING POINT STATUS
001750 172467 177754      LDF      SET7,ACC      :PRELOAD 000000,000000 INTO ACC
001754 177467 177754      FPI7: LDCDF DAT7,ACC    :LOAD-CONVERT 100177,177777,177777,177777
001760 170200      STFPS   FPS           :STORE FLOATING POINT STATUS
001762 170367 177034      STST    FEC           :STORE EXCEPTION CODES
001766 022700 147414      CMP      #147414,FPS   :CHECK FLOATING POINT STATUS
001772 001401      BEQ      .+4            :BRANCH IF OK
001774 104000      HLT                        :FPS NOT EQUAL TO 147414

001776 022767 000014 177016      CMP      #14,FEC       :CHECK FLOATING EXCEPTION CODE
002004 001401      BEQ      .+4            :BRANCH IF OK
002006 104000      HLT                        :FEC NOT EQUAL TO 14

002010 022767 001754 177006      CMP      #FPI7,FEA     :CHECK FLOATING EXCEPTION ADDRESS
002016 001401      BEQ      .+4            :BRANCH IF OK
002020 104000      HLT                        :FEA NOT EQUAL TO FPI7

002022 174067 176754      STF      ACC,ANS1      :STORE ACC IN ANS1,ANS2
002026 022767 000000 176746      CMP      #000000,ANS1   :DID 000000 GET STORED?
002034 001401      BEQ      .+4            :BRANCH IF OK
002036 104002      HLT+2                    :ANS1 NOT EQUAL TO 000000

002040 022767 000000 176736      CMP      #000000,ANS2   :DID 000000 GET STORED?
002046 001401      BEQ      .+4            :BRANCH IF OK
002050 104002      HLT+2                    :ANS2 NOT EQUAL TO 000000

```

```

:*****

```

:TEST 10: TEST LDCDF (LOAD-CONVERT DOUBLE TO FLOATING)
: LOAD 040000,000000,000000,000000 --> 040000,000000
: FPS = 047400, SRC = M6-R7, AC = AC2
:*****

002052 104400
002054 000404

SCOPE
BR TST10

002056 040000 000000 000000 DAT10: 040000,000000,000000,000000
002064 000000

002066 170127 047417
002072 177667 177760
002076 170200
002100 022700 047400
002104 001401
002106 104000

TST10: LDFPS #047417
FPI10: LDCDF DAT10, AC2
STFPS FPS
CMP #047400,FPS
BEQ .+4
HLT

:LOAD FLOATING POINT STAU
:LOAD-CONVERT 040000,000000,000000,000000
:STORE FLOATING POINT STATUS
:CHECK FLOATING POINT STATUS
:BRANCH IF OK
:FPS NOT EQUAL TO 047400

002110 174267 176666
002114 022767 040000 176660
002122 001401
002124 104002

STF AC2 ANS1
CMP #040000,ANS1
BEQ .+4
HLT+2

:STORE AC2 IN ANS1, ANS2
:DID 040000 GET STORED?
:BRANCH IF OK
:ANS1 NOT EQUAL TO 040000

002126 022767 000000 176650
002134 001401
002136 104002

CMP #000000,ANS2
BEQ .+4
HLT+2

:DID 000000 GET STORED?
:BRANCH IF OK
:ANS2 NOT EQUAL TO 000000

:*****
:TEST 11: TEST LDCDF (LOAD-CONVERT DOUBLE TO FLOATING)
: LOAD 140000,000000,100000,000000 --> 140000,000001
: FPS = 047410, SRC = M6-R7, AC = AC3
:*****

002140 104400
002142 000404

SCOPE
BR TST11

002144 140000 000000 100000 DAT11: 140000,000000,100000,000000
002152 000000

002154 170127 047417
002160 177767 177760
002164 170200
002166 022700 047410
002172 001401
002174 104000

TST11: LDFPS #047417
FPI11: LDCDF DAT11, AC3
STFPS FPS
CMP #047410,FPS
BEQ .+4
HLT

:LOAD FLOATING POINT STAU
:LOAD-CONVERT 140000,000000,100000,000000
:STORE FLOATING POINT STATUS
:CHECK FLOATING POINT STATUS
:BRANCH IF OK
:FPS NOT EQUAL TO 047410

002176 174367 176600
002202 022767 140000 176572
002210 001401
002212 104002

STF AC3 ANS1
CMP #140000,ANS1
BEQ .+4
HLT+2

:STORE AC3 IN ANS1, ANS2
:DID 140000 GET STORED?
:BRANCH IF OK
:ANS1 NOT EQUAL TO 140000

002214 022767 000001 176562
002222 001401
002224 104002

CMP #000001,ANS2
BEQ .+4
HLT+2

:DID 000001 GET STORED?
:BRANCH IF OK
:ANS2 NOT EQUAL TO 000001

:TEST 12: TEST LDCDF (LOAD-CONVERT DOUBLE TO FLOATING)
:LOAD 177777,177777,077777,177777 --> 177777,177777
:FPS = 047410, SRC = M6-R7, AC = AC1
:*****

```
002226 104400          SCOPE
002230 000404          BR      TST12

002232 177777 177777 077777 DAT12: 177777,177777,077777,177777
002240 177777

002242 170127 047417          TST12: LDFPS #047417          :LOAD FLOATING POINT STAUS
002246 177567 177760          FPI12: LDCDF  DAT12, AC1      :LOAD-CONVERT 177777,177777,077777,177777
002252 170200          STFPS FPS          :STORE FLOATING POINT STATUS
002254 022700 047410          CMP #047410,FPS      :CHECK FLOATING POINT STATUS
002260 001401          BEQ .+4            :BRANCH IF OK
002262 104000          HLT              :FPS NOT EQUAL TO 047410

002264 174167 176512          STF AC1, ANS1      :STORE AC1 IN ANS1, ANS2
002270 022767 177777 176504          CMP #177777,ANS1   :DID 177777 GET STORED?
002276 001401          BEQ .+4            :BRANCH IF OK
002300 104002          HLT+2            :ANS1 NOT EQUAL TO 177777

002302 022767 177777 176474          CMP #177777,ANS2   :DID 177777 GET STORED?
002310 001401          BEQ .+4            :BRANCH IF OK
002312 104002          HLT+2            :ANS2 NOT EQUAL TO 177777
```

:TEST 13: TEST LDCDF (LOAD-CONVERT DOUBLE TO FLOATING)
:LOAD 177777,177777,177777,177777 --> 100000,000000
:FPS = 147416, SRC = M6-R7, AC = AC1
:FEC = 10, FEA = FPI13
:*****

```
002314 104400          SCOPE
002316 000404          BR      TST13

002320 177777 177777 177777 DAT13: 177777,177777,177777,177777
002326 177777

002330 170127 047417          TST13: LDFPS #047417          :LOAD FLOATING POINT STAUS
002334 177567 177760          FPI13: LDCDF  DAT13, AC1      :LOAD-CONVERT 177777,177777,177777,177777
002340 170200          STFPS FPS          :STORE FLOATING POINT STATUS
002342 170367 176454          STST FEC          :STORE EXCEPTION CODES
002346 022700 147416          CMP #147416,FPS      :CHECK FLOATING POINT STATUS
002352 001401          BEQ .+4            :BRANCH IF OK
002354 104000          HLT              :FPS NOT EQUAL TO 147416

002356 022767 000010 176436          CMP #10, FEC       :CHECK FLOATING EXCEPTION CODE
002364 001401          BEQ .+4            :BRANCH IF OK
002366 104000          HLT              :FEC NOT EQUAL TO 10

002370 022767 002334 176426          CMP #FPI13, FEA    :CHECK FLOATING EXCEPTION ADDRESS
```

E02

MAINDEC-11-DCFPI-9
DCFPI.P11

TEST OF LDCDF, LDCFD, STCFD, STCDF
TEST SECTION

MACY11 27(732) 17-SEP-76 10:46 PAGE 17

```

002376 001401      BEQ      .+4      ;BRANCH IF OK
002400 104000      HLT                    ;FEA NOT EQUAL TO FPI13

002402 174167 176374  STF      AC1, ANS1  ;STORE AC1 IN ANS1, ANS2
002406 022767 100000 176366  CMP      #100000,ANS1 ;DID 100000 GET STORED?
002414 001401      BEQ      .+4      ;BRANCH IF OK
002416 104002      HLT+2           ;ANS1 NOT EQUAL TO 100000

002420 022767 000000 176356  CMP      #000000,ANS2 ;DID 000000 GET STORED?
002426 001401      BEQ      .+4      ;BRANCH IF OK
002430 104002      HLT+2           ;ANS2 NOT EQUAL TO 000000

```

```

:*****
:TEST 14:      TEST LDCDF (LOAD-CONVERT DOUBLE TO FLOATING)
:LOAD      077777,177777,177777,177777 --> 000000,000000
:FPS = 147406, SRC = M6-R7, AC = AC3
:FEC = 10,   FEA = FPI14
:*****

```

```

002432 104400      SCOPE
002434 000404      BR      TST14

002436 077777 177777 177777  DAT14: 077777,177777,177777,177777
002444 177777

002446 170127 047417  TST14: LDFPS  #047417 ;LOAD FLOATING POINT STATUS -
002452 177767 177760  FPI14: LDCDF  DAT14, AC3 ;LOAD-CONVERT 077777,177777,177777,177777
002456 170200      STFPS  FPS ;STORE FLOATING POINT STATUS
002460 170367 176336  STST   FEC ;STORE EXCEPTION CODES
002464 022700 147406  CMP    #147406,FPS ;CHECK FLOATING POINT STATUS
002470 001401      BEQ    .+4 ;BRANCH IF OK
002472 104000      HLT                    ;FPS NOT EQUAL TO 147406

002474 022767 000010 176320  CMP    #10, FEC ;CHECK FLOATING EXCEPTION CODE
002502 001401      BEQ    .+4 ;BRANCH IF OK
002504 104000      HLT                    ;FEC NOT EQUAL TO 10

002506 022767 002452 176310  CMP    #FPI14, FEA ;CHECK FLOATING EXCEPTION ADDRESS
002514 001401      BEQ    .+4 ;BRANCH IF OK
002516 104000      HLT                    ;FEA NOT EQUAL TO FPI14

002520 174367 176256  STF    AC3, ANS1 ;STORE AC3 IN ANS1, ANS2
002524 022767 000000 176250  CMP    #000000,ANS1 ;DID 000000 GET STORED?
002532 001401      BEQ    .+4 ;BRANCH IF OK
002534 104002      HLT+2           ;ANS1 NOT EQUAL TO 000000

002536 022767 000000 176240  CMP    #000000,ANS2 ;DID 000000 GET STORED?
002544 001401      BEQ    .+4 ;BRANCH IF OK
002546 104002      HLT+2           ;ANS2 NOT EQUAL TO 000000

```

```

:*****
:TEST 15:      TEST LDCDF (LOAD-CONVERT DOUBLE TO FLOATING)
:LOAD      040252,125252,125252,125252 --> 040252,125253
:FPS = 047400, SRC = M0-AC1, AC = AC2
:*****

```

```

002550 104400          SCOPE
002552 000404          BR      TST15

002554 040252 125252 125252 DAT15: 040252,125252,125252,125252
002556 125252

002564 170127 047417      TST15: LDFPS  #047417      ;LOAD FLOATING POINT STAUS
002570 170011          SETD
002572 172667 177756      LDD      DAT15, AC2      ;
002576 170001          SETF
002600 177502          LDCDF  AC2,   AC1      ;LOAD-CONVERT 040252,125252,125252,125252
002602 170200          STFPS  FPS
002604 022700 047400      CMP      #047400,FPS    ;STORE FLOATING POINT STATUS
002610 001401          BEQ      .+4          ;CHECK FLOATING POINT STATUS
002612 104000          HLT      ;BRANCH IF OK
                                ;FPS NOT EQUAL TO 047400

002614 174167 176162      STF      AC1,   ANS1    ;STORE AC1 IN ANS1, ANS2
002620 022767 040252 176154 CMP      #040252,ANS1  ;DID 040252 GET STORED?
002626 001401          BEQ      .+4          ;BRANCH IF OK
002630 104002          HLT+2     ;ANS1 NOT EQUAL TO 040252

002632 022767 125253 176144 CMP      #125253,ANS2  ;DID 125253 GET STORED?
002640 001401          BEQ      .+4          ;BRANCH IF OK
002642 104002          HLT+2     ;ANS2 NOT EQUAL TO 125253

```

```

*****
:TEST 16:      TEST LDCDF (LOAD-CONVERT DOUBLE TO FLOATING)
:              LOAD  140125,052525,052525,052525 --> 140125,052525
:              FPS = 047410,   SRC = M1-R1,   AC = AC3
*****

```

```

002644 104400          SCOPE
002646 000404          BR      TST16

002650 140125 052525 052525 DAT16: 140125,052525,052525,052525
002656 052525

002660 170127 047417      TST16: LDFPS  #047417      ;LOAD FLOATING POINT STAUS
002664 012701 002650      MOV      #DAT16, R1
002670 177711          LDCDF  (R1),  AC3    ;LOAD-CONVERT 140125,052525,052525,052525
002672 170200          STFPS  FPS
002674 022700 047410      CMP      #047410,FPS    ;STORE FLOATING POINT STATUS
002700 001401          BEQ      .+4          ;CHECK FLOATING POINT STATUS
002702 104000          HLT      ;BRANCH IF OK
                                ;FPS NOT EQUAL TO 047410

002704 174367 176072      STF      AC3,   ANS1    ;STORE AC3 IN ANS1, ANS2
002710 022701 002650      CMP      #DAT16, R1    ;CHECK R1
002714 000401          BR      .+4          ;BRANCH IF OK
002716 104000          HLT      ;R1 NOT EQUAL TO #DAT16

002720 022767 140125 176054 CMP      #140125,ANS1  ;DID 140125 GET STORED?
002726 001401          BEQ      .+4          ;BRANCH IF OK
002730 104002          HLT+2     ;ANS1 NOT EQUAL TO 140125

```

```

002732 022767 052525 176044      CMP      #052525,ANS2      ;DID 052525 GET STORED?
002740 001401                      BEQ      .+4              ;BRANCH IF OK
002742 104002                      HLT+2                    ;ANS2 NOT EQUAL TO 052525

```

```

:*****
:TEST 17:      TEST LDCDF (LOAD-CONVERT DOUBLE TO FLOATING)
:              LOAD      040125,052525,052525,052525 --> 040125,052525
:              FPS = 047400, SRC = M2-R5, AC = AC1
:*****

```

```

002744 104400
002746 000404

```

```

SCOPE
BR      TST17

```

```

002750 040125 052525 052525 DAT17: 040125,052525,052525,052525
002756 052525

```

```

002760 170127 047417      TST17: LDFPS      #047417      ;LOAD FLOATING POINT STAUS
002764 012705 002750      MOV      #DAT17, R5
002770 177525                      LDCDF    (R5)+, AC1      ;LOAD-CONVERT 040125,052525,052525,052525
002772 170200                      STFPS    FPS              ;STORE FLOATING POINT STATUS
002774 022700 047400      CMP      #047400,FPS     ;CHECK FLOATING POINT STATUS
003000 001401                      BEQ      .+4              ;BRANCH IF OK
003002 104000                      HLT

```

```

003004 174167 175772      STF      AC1, ANS1      ;STORE AC1 IN ANS1, ANS2
003010 022705 002760      CMP      #TST17, R5    ;CHECK R5
003014 001401                      BEQ      .+4              ;BRANCH IF OK
003016 104000                      HLT                      ;R5 NOT EQUAL TO #TST17

```

```

003020 022767 040125 175754      CMP      #040125,ANS1   ;DID 040125 GET STORED?
003026 001401                      BEQ      .+4              ;BRANCH IF OK
003030 104002                      HLT+2                    ;ANS1 NOT EQUAL TO 040125

```

```

003032 022767 052525 175744      CMP      #052525,ANS2   ;DID 052525 GET STORED?
003040 001401                      BEQ      .+4              ;BRANCH IF OK
003042 104002                      HLT+2                    ;ANS2 NOT EQUAL TO 052525

```

```

:*****
:TEST 20:      TEST LDCDF (LOAD-CONVERT DOUBLE TO FLOATING)
:              LOAD      140252,125252,125252,125252 --> 140252,125253
:              FPS = 047410, SRC = M3-R6, AC = AC2
:*****

```

```

003044 104400
003046 000405

```

```

SCOPE
BR      TST20

```

```

003050 140252 125252 125252 DAT20: 140252,125252,125252,125252
003056 125252
003060 000000 SAV20: 0

```

```

003062 170127 047417      TST20: LDFPS      #047417      ;LOAD FLOATING POINT STAUS
003066 010667 177766      MOV      %6, SAV20      ;SAVE THE STACK POINTER
003072 012746 003050      MOV      #DAT20, -(%6)  ;PUT #DAT20 ON THE STACK

```

```

003076 177636          LDCDF  2(%6)+, AC2    ;LOAD-CONVERT 140252,125252,125252,125252
003100 170200          STFPS  FPS          ;STORE FLOATING POINT STATUS
003102 020667 177752  CMP     %6, SAV20   ;CHECK THE STACK POINTER
003106 001405          BEQ     TSA20      ;BRANCH IF OK
003110 010667 175666  MOV     %6, ANS1    ;SAVE WRONG %6 FOR TYPING
003114 016706 177740  MOV     SAV20, %6   ;RESTORE STACK POINTER
003120 104001          HLT+1   ;STACK POINTER FOULED UP, NOT 500?

003122 020027 047410  TSA20:  CMP     FPS, #047410 ;CHECK FLOATING POINT STATUS
003126 001401          BEQ     .+4         ;BRANCH IF OK
003130 104000          HLT     ;FLOATING POINT STATUS NOT 047410

003132 174267 175644  STF     AC2, ANS1   ;STORE AC2 IN ANS1, ANS2
003136 022767 140252 175636  CMP     #140252,ANS1 ;DID 140252 GET STORED?
003144 001401          BEQ     .+4         ;BRANCH IF OK
003146 104002          HLT+2   ;ANS1 NOT EQUAL TO 140252

003150 022767 125253 175626  CMP     #125253,ANS2 ;DID 125253 GET STORED?
003156 001401          BEQ     .+4         ;BRANCH IF OK
003160 104002          HLT+2   ;ANS2 NOT EQUAL TO 125253

```

```

:*****
:TEST 21: TEST LDCDF (LOAD-CONVERT DOUBLE TO FLOATING)
:LOAD 052525,052525,052525,052525 --> 052525,052525
:FPS = 047400, SRC = M4-R1, AC = ACC
:*****

```

```

003162 104400          SCOPE
003164 000404          BR     TST21

003166 052525 052525 052525 052525  DAT21: 052525,052525,052525,052525

003176 170127 047417  TST21:  LDFPS  #047417   ;LOAD FLOATING POINT STAUS
003202 012701 003176  MOV     #TST21, R1  ;LOAD R1
003206 177441          LDCDF  -(R1), ACC  ;LOAD-CONVERT 052525,052525,052525,052525
003210 170200          STFPS  FPS          ;STORE FLOATING POINT STATUS
003212 022700 047400  CMP     #047400,FPS ;CHECK FLOATING POINT STATUS
003216 001401          BEQ     .+4         ;BRANCH IF OK
003220 104000          HLT     ;FPS NOT EQUAL TO 047400

003222 174067 175554  STF     ACC, ANS1   ;STORE ACC IN ANS1, ANS2
003226 022701 003166  CMP     #DAT21, R1  ;CHECK R1
003232 001401          BEQ     .+4         ;BRANCH IF OK
003234 104000          HLT     ;R1 NOT EQUAL TO #DAT21

003236 022767 052525 175536  CMP     #052525,ANS1 ;DID 052525 GET STORED?
003244 001401          BEQ     .+4         ;BRANCH IF OK
003246 104002          HLT+2   ;ANS1 NOT EQUAL TO 052525

003250 022767 052525 175526  CMP     #052525,ANS2 ;DID 052525 GET STORED?
003256 001401          BEQ     .+4         ;BRANCH IF OK
003260 104002          HLT+2   ;ANS2 NOT EQUAL TO 052525

```

```

*****
:TEST 22: TEST LDCDF (LOAD-CONVERT DOUBLE TO FLOATING)
:LOAD 125252,125252,125252,125252 --> 125252,125253
:FPS = 047410, SRC = M5-R2, AC = AC3
*****

```

```

003262 104400 SCOPE
003264 000405 BR TST22

003266 125252 125252 125252 DAT22: 125252,125252,125252,125252
003274 125252 ADR22: DAT22
003276 003266

003300 170127 047417 TST22: LDFPS #047417 ;LOAD FLOATING POINT STAUS
003304 012702 003300 MOV #TST22, R2 ;LOAD R2
003310 177752 LDCDF @-(R2), AC3 ;LOAD-CONVERT 125252,125252,125252,125252
003312 170200 STFPS FPS ;STORE FLOATING POINT STATUS
003314 022700 047410 CMP #047410,FPS ;CHECK FLOATING POINT STATUS
003320 001401 BEQ .+4 ;BRANCH IF OK
003322 104000 HLT ;FPS NOT EQUAL TO 047410

003324 174367 175452 STF AC3, ANS1 ;STORE AC3 IN ANS1, ANS2
003330 022702 003276 CMP #ADR22, R2 ;CHECK R2
003334 001401 BEQ .+4 ;BRANCH IF OK
003336 104000 HLT ;R2 NOT EQUAL TO #ADR22

003340 022767 125252 175434 CMP #125252,ANS1 ;DID 125252 GET STORED?
003346 001401 BEQ .+4 ;BRANCH IF OK
003350 104002 HLT+2 ;ANS1 NOT EQUAL TO 125252

003352 022767 125253 175424 CMP #125253,ANS2 ;DID 125253 GET STORED?
003360 001401 BEQ .+4 ;BRANCH IF OK
003362 104002 HLT+2 ;ANS2 NOT EQUAL TO 125253

```

```

*****
:TEST 23: TEST LDCDF (LOAD-CONVERT DOUBLE TO FLOATING)
:LOAD 040252,125252,125252,125252 --> 040252,125253
:FPS = 047400, SRC = M6-R0, AC = AC0
*****

```

```

003364 104400 SCOPE
003366 000405 BR TST23

003370 040252 125252 125252 DAT23: 040252,125252,125252,125252
003376 125252 SAV23: 0
003400 000000

003402 170127 047417 TST23: LDFPS #047417 ;LOAD FLOATING POINT STAUS
003406 012700 004026 MOV #DAT23+436,R0 ;LOAD-CONVERT 040252,125252,125252,125252
003412 177460 177342 LDCDF -436(R0),AC0 ;SAVE R0
003416 010067 177756 MOV R0, SAV23 ;STORE FLOATING POINT STATUS
003422 170200 STFPS FPS ;CHECK FLOATING POINT STATUS
003424 022700 047400 CMP #047400,FPS ;BRANCH IF OK
003430 001401 BEQ .+4 ;FPS NOT EQUAL TO 047400
003432 104000 HLT

```

```

003434 174067 175342           STF      ACO      ANS1      ;STORE ACO IN ANS1, ANS2
003440 022767 004026 177732     CMP      #DAT23+436,SAV23 ;CHECK SAVED RO
003446 001401           BEQ      .+4             ;BRANCH IF OK
003450 104000           HLT                               ;RO NOT EQUAL TO #DAT23+436

003452 022767 040252 175322     CMP      #040252,ANS1    ;DID 040252 GET STORED?
003460 001401           BEQ      .+4             ;BRANCH IF OK
003462 104002           HLT+2                   ;ANS1 NOT EQUAL TO 040252

003464 022767 125253 175312     CMP      #125253,ANS2    ;DID 125253 GET STORED?
003472 001401           BEQ      .+4             ;BRANCH IF OK
003474 104002           HLT+2                   ;ANS2 NOT EQUAL TO 125253

```

```

:*****
:TEST 24:      TEST LDCDF (LOAD-CONVERT DOUBLE TO FLOATING)
:              LOAD      140125,052525,052525,052525 --> 140125,052525
:              FPS = 047410, SRC = M7-R5, AC = AC1
:*****

```

```

003476 104400           SCOPE
003500 000405           BR      TST24

003502 140125 052525 052525 DAT24: 140125,052525,052525,052525
003510 052525

003512 003502           ADR24: DAT24
003514 170127 047417           TST24: LDFPS      #047417      ;LOAD FLOATING POINT STATUS
003520 012705 003133           MOV      #ADR24-357,R5   ;LOAD R5
003524 177575 000357           LDCDF   3357(R5),AC1    ;LOAD-CONVERT 140125,052525,052525,052525
003530 170200           STFPS   FPS             ;STORE FLOATING POINT STATUS
003532 022700 047410           CMP      #047410,FPS     ;CHECK FLOATING POINT STATUS
003536 001401           BEQ      .+4             ;BRANCH IF OK
003540 104000           HLT                               ;FPS NOT EQUAL TO 047410

003542 174167 175234           STF      AC1      ANS1    ;STORE AC1 IN ANS1, ANS2
003546 022705 003133           CMP      #ADR24-357,R5   ;CHECK R5
003552 001401           BEQ      .+4             ;BRANCH IF OK
003554 104000           HLT                               ;R5 NOT EQUAL TO #ADR24-357

003556 022767 140125 175216     CMP      #140125,ANS1    ;DID 140125 GET STORED?
003564 001401           BEQ      .+4             ;BRANCH IF OK
003566 104002           HLT+2                   ;ANS1 NOT EQUAL TO 140125

003570 022767 052525 175206     CMP      #052525,ANS2    ;DID 052525 GET STORED?
003576 001401           BEQ      .+4             ;BRANCH IF OK
003600 104002           HLT+2                   ;ANS2 NOT EQUAL TO 052525

```

```

:*****
:TEST 25:      TEST LDCDF (LOAD-CONVERT DOUBLE TO FLOATING)
:              +0.001804 GETS ASSEMBLED AS 035754
:              LOAD      035754 --> 035754,000000
:              FPS = 047400, SRC = M2-R7, AC = AC2
:*****

```

```

003602 104400          SCOPE
003604 170127 047417   TST25: LDFPS #047417      ;LOAD FLOATING POINT STAU
003610 177627 035754   LDCDF #+0.001804,AC2 ;LOAD-CONVERT #+0.001804
003614 170200          STFPS FPS      ;STORE FLOATING POINT STATUS
003616 022700 047400   CMP #047400,FPS    ;CHECK FLOATING POINT STATUS
003622 001401          BEQ .+4      ;BRANCH IF OK
003624 104000          HLT                    ;FPS NOT EQUAL TO 047400

003626 174267 175150   STF AC2, ANS1     ;STORE AC2 IN ANS1, ANS2
003632 022767 035754 175142 CMP #035754,ANS1 ;DID 035754 GET STORED?
003640 001401          BEQ .+4      ;BRANCH IF OK
003642 104002          HLT+2          ;ANS1 NOT EQUAL TO 035754

003644 005767 175134   TST ANS2         ;DID ZERO GET STORED?
003650 001401          BEQ .+4      ;BRANCH IF OK
003652 104002          HLT+2          ;ANS2 NOT EQUAL TO ZERO

```

```

*****
:TEST 26:      TEST LDCDF (LOAD-CONVERT DOUBLE TO FLOATING)
:              LOAD 052525,052525,052525,052525 --> 052525,052525
:              FPS = 047400, SRC = M3-R7, AC = AC1
*****

```

```

003654 104400          SCOPE
003656 000404          BR      TST26

003660 052525 052525 052525 DAT26: 052525,052525,052525,052525
003666 052525

003670 170127 047417   TST26: LDFPS #047417      ;LOAD FLOATING POINT STAU
003674 177537 003660   LDCDF @DAT26,AC1 ;LOAD-CONVERT 052525,052525,052525,052525
003700 170200          STFPS FPS      ;STORE FLOATING POINT STATUS
003702 022700 047400   CMP #047400,FPS    ;CHECK FLOATING POINT STATUS
003706 001401          BEQ .+4      ;BRANCH IF OK
003710 104000          HLT                    ;FPS NOT EQUAL TO 047400

003712 174167 175064   STF AC1, ANS1     ;STORE AC1 IN ANS1, ANS2
003716 022767 052525 175056 CMP #052525,ANS1 ;DID 052525 GET STORED?
003724 001401          BEQ .+4      ;BRANCH IF OK
003726 104002          HLT+2          ;ANS1 NOT EQUAL TO 052525

003730 022767 052525 175046 CMP #052525,ANS2 ;DID 052525 GET STORED?
003736 001401          BEQ .+4      ;BRANCH IF OK
003740 104002          HLT+2          ;ANS2 NOT EQUAL TO 052525

```

```

*****
:TEST 27:      TEST LDCDF (LOAD-CONVERT DOUBLE TO FLOATING)
:              LOAD 125252,125252,125252,125252 --> 125252,125253
:              FPS = 047410, SRC = M7-R7, AC = AC3
*****

```

```

003742 104400          SCOPE

```



```

003744 000405          BR      TST27
003746 125252 125252 125252 125252  DAT27: 125252,125252,125252,125252
003754 125252
003756 003746          ADR27: DAT27
003760 170127 047417          TST27: LDFPS  #047417          ;LOAD FLOATING POINT STAU
003764 177777 177766          LDCDF  QADR27, AC3      ;LOAD-CONVERT 125252,125252,125252,125252
003770 170200          STFPS  FPS              ;STORE FLOATING POINT STATUS
003772 022700 047410          CMP    #047410,FPS      ;CHECK FLOATING POINT STATUS
003776 001401          BEQ    .+4              ;BRANCH IF OK
004000 104000          HLT                    ;FPS NOT EQUAL TO 047410
004002 174367 174774          STF    AC3, ANS1        ;STORE AC3 IN ANS1, ANS2
004006 022767 125252 174766          CMP    #125252,ANS1     ;DID 125252 GET STORED?
004014 001401          BEQ    .+4              ;BRANCH IF OK
004016 104002          HLT+2                  ;ANS1 NOT EQUAL TO 125252
004020 022767 125253 174756          CMP    #125253,ANS2     ;DID 125253 GET STORED?
004026 001401          BEQ    .+4              ;BRANCH IF OK
004030 104002          HLT+2                  ;ANS2 NOT EQUAL TO 125253

```

```

:*****
:TEST 30: TEST LDCFD (LOAD-CONVERT FLOATING TO DOUBLE)
:LOAD 000000,000000 --> 000000,000000,000000,000000
:FPS = 047604, SRC = M6-R7, AC = AC3
:*****

```

```

004032 104400          SCOPE
004034 000402          BR      TST30
004036 000000 000000          DAT30: 000000,000000
004042 170127 047617          TST30: LDFPS  #047617          ;LOAD FLOATING POINT STAU
004046 172704          LDD    AC4, AC3        ;PRELOAD AC3 WITH GARBAGE
004050 177767 177762          FPI30: LDCFD  DAT30, AC3      ;LOAD-CONVERT 000000,000000 INTO AC3
004054 170200          STFPS  FPS              ;STORE FLOATING POINT STATUS
004056 022700 047604          CMP    #047604,FPS      ;CHECK FLOATING POINT STATUS
004062 001401          BEQ    .+4              ;BRANCH IF OK
004064 104000          HLT                    ;FPS NOT EQUAL TO 047604
004066 174367 174710          STD    AC3, ANS1        ;STORE AC3 IN ANS1 THRU ANS4
004072 005767 174704          TST    ANS1              ;DID ZERO GET STORED?
004076 001401          BEQ    .+4              ;BRANCH IF OK
004100 104004          HLT+4                  ;ANS1 NOT EQUAL TO ZERO
004102 005767 174676          TST    ANS2              ;DID ZERO GET STORED?
004106 001401          BEQ    .+4              ;BRANCH IF OK
004110 104004          HLT+4                  ;ANS2 NOT EQUAL TO ZERO
004112 005767 174670          TST    ANS3              ;DID ZERO GET STORED?
004116 001401          BEQ    .+4              ;BRANCH IF OK
004120 104004          HLT+4                  ;ANS3 NOT EQUAL TO ZERO
004122 005767 174662          TST    ANS4              ;DID ZERO GET STORED?

```

004126 001401
004130 104004

BEQ .+4
HLT+4

;BRANCH IF OK
;ANS4 NOT EQUAL TO ZERO

:TEST 31: TEST LDCFD (LOAD-CONVERT FLOATING TO DOUBLE)
:LOAD 052525,052525 --> 052525,052525,000000,000000
:FPS = 047600, SRC = M6-R7, AC = AC1
:*****

004132 104400
004134 000402

SCOPE
BR TST31

004136 052525 052525

DAT31: 052525,052525

004142 170127 047617
004146 172504
004150 177567 177762
004154 170200
004156 022700 047600
004162 001401
004164 104000

TST31: LDFPS #047617
LDD AC4, AC1
FPI31: LDCFD DAT31, AC1
STFPS FPS
CMP #047600,FPS
BEQ .+4
HLT

;LOAD FLOATING POINT STAU
;PRELOAD AC1 WITH GARBAGE
;LOAD-CONVERT 052525,052525 INTO AC1
;STORE FLOATING POINT STATUS
;CHECK FLOATING POINT STATUS
;BRANCH IF OK
;FPS NOT EQUAL TO 047600

004166 174167 174610
004172 022767 052525 174602
004200 001401
004202 104004

STD AC1, ANS1
CMP #052525,ANS1
BEQ .+4
HLT+4

;STORE AC1 IN ANS1 THRU ANS4
;DID 052525 GET STORED?
;BRANCH IF OK
;ANS1 NOT EQUAL TO 052525

004204 022767 052525 174572
004212 001401
004214 104004

CMP #052525,ANS2
BEQ .+4
HLT+4

;DID 052525 GET STORED?
;BRANCH IF OK
;ANS2 NOT EQUAL TO 052525

004216 005767 174564
004222 001401
004224 104004

TST ANS3
BEQ .+4
HLT+4

;DID ZERO GET STORED?
;BRANCH IF OK
;ANS3 NOT EQUAL TO ZERO

004226 005767 174556
004232 001401
004234 104004

TST ANS4
BEQ .+4
HLT+4

;DID ZERO GET STORED?
;BRANCH IF OK
;ANS4 NOT EQUAL TO ZERO

:TEST 32: TEST LDCFD (LOAD-CONVERT FLOATING TO DOUBLE)
:LOAD 125252,125252 --> 125252,125252,000000,000000
:FPS = 047610, SRC = M6-R7, AC = AC0
:*****

004236 104400
004240 000402

SCOPE
BR TST32

004242 125252 125252

DAT32: 125252,125252

004246 170127 047617
004252 172404
004254 177467 177762
004260 170200

TST32: LDFPS #047617
LDD AC4, AC0
FPI32: LDCFD DAT32, AC0
STFPS FPS

;LOAD FLOATING POINT STAU
;PRELOAD AC0 WITH GARBAGE
;LOAD-CONVERT 125252,125252 INTO AC0
;STORE FLOATING POINT STATUS

```

004262 022700 047610      CMP      #047610,FPS      ;CHECK FLOATING POINT STATUS
004266 001401      BEQ      .+4          ;BRANCH IF OK
004270 104000      HLT                      ;FPS NOT EQUAL TO 047610

004272 174067 174504      STD      AC0, ANS1     ;STORE AC0 IN ANS1 THRU ANS4
004276 022767 125252 174476    CMP      #125252,ANS1 ;DID 125252 GET STORED?
004304 001401      BEQ      .+4          ;BRANCH IF OK
004306 104004      HLT+4        ;ANS1 NOT EQUAL TO 125252

004310 022767 125252 174466    CMP      #125252,ANS2 ;DID 125252 GET STORED?
004316 001401      BEQ      .+4          ;BRANCH IF OK
004320 104004      HLT+4        ;ANS2 NOT EQUAL TO 125252

004322 005767 174460      TST      ANS3         ;DID ZERO GET STORED?
004326 001401      BEQ      .+4          ;BRANCH IF OK
004330 104004      HLT+4        ;ANS3 NOT EQUAL TO ZERO

004332 005767 174452      TST      ANS4         ;DID ZERO GET STORED?
004336 001401      BEQ      .+4          ;BRANCH IF OK
004340 104004      HLT+4        ;ANS4 NOT EQUAL TO ZERO

```

```

*****
:TEST 33:      TEST LDCFD (LOAD-CONVERT FLOATING TO DOUBLE)
:              LOAD 000200,000000 --> 000200,000000,000000,000000
:              FPS = 047600, SRC = M6-R7, AC = AC3
*****

```

```

004342 104400      SCOPE
004344 000402      BR      TST33

004346 000200 000000      DAT33: 000200,000000

004352 170127 047617      TST33: LDFPS #047617      ;LOAD FLOATING POINT STAUS
004356 172704      LDD      AC4, AC3     ;PRELOAD AC3 WITH GARBAGE
004360 177767 177762      FPI33: LDCFD DAT33, AC3 ;LOAD-CONVERT 000200,000000 INTO AC3
004364 170200      STFPS   FPS          ;STORE FLOATING POINT STATUS
004366 022700 047600      CMP      #047600,FPS ;CHECK FLOATING POINT STATUS
004372 001401      BEQ      .+4          ;BRANCH IF OK
004374 104000      HLT                      ;FPS NOT EQUAL TO 047600

004376 174367 174400      STD      AC3, ANS1     ;STORE AC3 IN ANS1 THRU ANS4
004402 022767 000200 174372    CMP      #000200,ANS1 ;DID 000200 GET STORED?
004410 001401      BEQ      .+4          ;BRANCH IF OK
004412 104004      HLT+4        ;ANS1 NOT EQUAL TO 000200

004414 022767 000000 174362    CMP      #000000,ANS2 ;DID 000000 GET STORED?
004422 001401      BEQ      .+4          ;BRANCH IF OK
004424 104004      HLT+4        ;ANS2 NOT EQUAL TO 000000

004426 005767 174354      TST      ANS3         ;DID ZERO GET STORED?
004432 001401      BEQ      .+4          ;BRANCH IF OK
004434 104004      HLT+4        ;ANS3 NOT EQUAL TO ZERO

004436 005767 174346      TST      ANS4         ;DID ZERO GET STORED?
004442 001401      BEQ      .+4          ;BRANCH IF OK

```

MAINDEC-11-00FPI-8
00000000

TEST OF LDCFD, LDCFD, STCFD, STCFD
TEST SECTION

004444 104004

HLT+4

:ANS4 NOT EQUAL TO ZERO

```

*****
:TEST 34:      TEST LDCFD (LOAD-CONVERT FLOATING TO DOUBLE)
:LOAD      100200,000000 --> 100200,000000,000000,000000
:FPS = 047610, SRC = M6-R7, AC = AC2
*****

```

004446 104400
004450 000402

SCOPE
BR TST34

004452 100200 000000

DAT34: 100200,000000

004456 170127 047617
004462 172604
004464 177667 177762
004470 170200
004472 022700 047610
004476 001401
004500 104000

```

TST34: LDFPS #047617 :LOAD FLOATING POINT STATUS
LDD AC4 AC2 :PRELOAD AC2 WITH GARBAGE
FPI34: LDCFD DAT34, AC2 :LOAD-CONVERT 100200,000000 INTO AC2
STFPS FPS :STORE FLOATING POINT STATUS
CMP #047610,FPS :CHECK FLOATING POINT STATUS
BEQ .+4 :BRANCH IF OK
HLT :FPS NOT EQUAL TO 047610

```

004502 174267 174274
004506 022767 100200 174266
004514 001401
004516 104004

```

STD AC2, ANS1 :STORE AC2 IN ANS1 THRU ANS4
CMP #100200,ANS1 :DID 100200 GET STORED?
BEQ .+4 :BRANCH IF OK
HLT+4 :ANS1 NOT EQUAL TO 100200

```

004520 022767 000000 174256
004526 001401
004530 104004

```

CMP #000000,ANS2 :DID 000000 GET STORED?
BEQ .+4 :BRANCH IF OK
HLT+4 :ANS2 NOT EQUAL TO 000000

```

004532 005767 174250
004536 001401
004540 104004

```

TST ANS3 :DID ZERO GET STORED?
BEQ .+4 :BRANCH IF OK
HLT+4 :ANS3 NOT EQUAL TO ZERO

```

004542 005767 174242
004546 001401
004550 104004

```

TST ANS4 :DID ZERO GET STORED?
BEQ .+4 :BRANCH IF OK
HLT+4 :ANS4 NOT EQUAL TO ZERO

```

```

*****
:TEST 35:      TEST LDCFD (LOAD-CONVERT FLOATING TO DOUBLE)
:LOAD      000177,177777 --> 000000,000000,000000,000000
:FPS = 047604, SRC = M6-R7, AC = AC2
*****

```

004552 104400
004554 000402

SCOPE
BR TST35

004556 000177 177777

DAT35: 000177,177777

004562 170127 047617
004566 172604
004570 177667 177762
004574 170200
004576 022700 047604

```

TST35: LDFPS #047617 :LOAD FLOATING POINT STATUS
LDD AC4 AC2 :PRELOAD AC2 WITH GARBAGE
FPI35: LDCFD DAT35, AC2 :LOAD-CONVERT 000177,177777 INTO AC2
STFPS FPS :STORE FLOATING POINT STATUS
CMP #047604,FPS :CHECK FLOATING POINT STATUS

```

MAINDEC-11-DCFPI-8
DCFPI.P11

TEST OF LDCFD, LDCFD, STCFD, STCDF
TEST SECTION

```

004602 001401      BEQ      .+4      :BRANCH IF OK
004604 104000      HLT
004606 174267 174170  STD      AC2      ANS1  :STORE AC2 IN ANS1 THRU ANS4
004612 005767 174164  TST      ANS1     :DID ZERO GET STORED?
004616 001401      BEQ      .+4      :BRANCH IF OK
004620 104004      HLT+4     :ANS1 NOT EQUAL TO ZERO
004622 005767 174156  TST      ANS2     :DID ZERO GET STORED?
004626 001401      BEQ      .+4      :BRANCH IF OK
004630 104004      HLT+4     :ANS2 NOT EQUAL TO ZERO
004632 005767 174150  TST      ANS3     :DID ZERO GET STORED?
004636 001401      BEQ      .+4      :BRANCH IF OK
004640 104004      HLT+4     :ANS3 NOT EQUAL TO ZERO
004642 005767 174142  TST      ANS4     :DID ZERO GET STORED?
004646 001401      BEQ      .+4      :BRANCH IF OK
004650 104004      HLT+4     :ANS4 NOT EQUAL TO ZERO

```

```

*****
:TEST 36:      TEST LDCFD (LOAD-CONVERT FLOATING TO DOUBLE)
:              LOAD 100177,177777 --> 052525,055252,125252,125252
:              FPS = 147614, SRC = M6-R7, AC = AC3
:              FEC = 14, FEA = FPI36
*****

```

```

004652 104400      SCOPE
004654 000406      BR      TST36
004656 052525 055252 125252  SET36: 052525,055252,125252,125252
004664 125252
004666 100177 177777  DAT36: 100177,177777
004672 170127 047617  TST36: LDFPS #047617 :LOAD FLOATING POINT STATUS
004676 172767 177754  LDD      SET36, AC3 :LOAD 052525,055252,125252,125252 INTO AC3
004702 177767 177760  FPI36: LDCFD DAT36, AC3 :LOAD-CONVERT 100177,177777 INTO AC3
004706 170200      STFPS  FPS :STORE FLOATING POINT STATUS
004710 170367 174106  STST   FEC :STORE EXCEPTION CODES
004714 022700 147614  CMP     #147614,FPS :CHECK FLOATING POINT STATUS
004720 001401      BEQ     .+4      :BRANCH IF OK
004722 104000      HLT
004724 022767 000014 174070  CMP     #14, FEC :CHECK FLOATING EXCEPTION CODE
004732 001401      BEQ     .+4      :BRANCH IF OK
004734 104000      HLT
004736 022767 004702 174060  CMP     #FPI36, FEA :CHECK FLOATING EXCEPTION ADDRESS
004744 001401      BEQ     .+4      :BRANCH IF OK
004746 104000      HLT
004750 174367 174026  STD     AC3, ANS1 :STORE AC3 IN ANS1 THRU ANS4
004754 022767 052525 174020  CMP     #052525,ANS1 :DID 052525 GET STORED?
004762 001401      BEQ     .+4      :BRANCH IF OK
004764 104004      HLT+4     :ANS1 NOT EQUAL TO 052525

```

```

004766 022767 055252 174010      CMP      #055252,ANS2      :DID 055252 GET STORED?
004774 001401                      BEQ      .+4             :BRANCH IF OK
004776 104004                      HLT+4          :ANS2 NOT EQUAL TO 055252

005000 022767 125252 174000      CMP      #125252,ANS3      :DID 125252 GET STORED?
005006 001401                      BEQ      .+4             :BRANCH IF OK
005010 104004                      HLT+4          :ANS3 NOT EQUAL TO 125252

005012 022767 125252 173770      CMP      #125252,ANS4      :DID 125252 GET STORED?
005020 001401                      BEQ      .+4             :BRANCH IF OK
005022 104004                      HLT+4          :ANS4 NOT EQUAL TO 125252

```

```

:*****
:TEST 37:      TEST LDCFD (LOAD-CONVERT FLOATING TO DOUBLE)
:              LOAD  040000,000000 --> 040000,000000,000000,000000
:              FPS = 047600, SRC = M6-R7, AC = AC1
:*****

```

```

005024 104400      SCOPE
005026 000402      BR      TST37

005030 040000 000000      DAT37: 040000,000000

005034 170127 047617      TST37: LDFPS  #047617      :LOAD FLOATING POINT STATUS
005040 172504                      LDD      AC4, AC1        :PRELOAD AC1 WITH GARBAGE
005042 177567 177762      FPI37: LDCFD  DAT37, AC1    :LOAD-CONVERT 040000,000000 INTO AC1
005046 170200                      STFPS   FPS             :STORE FLOATING POINT STATUS
005050 022700 047600      CMP      #047600,FPS     :CHECK FLOATING POINT STATUS
005054 001401                      BEQ      .+4             :BRANCH IF OK
005056 104000                      HLT      :FPS NOT EQUAL TO 047600

005060 174167 173716      STD      AC1, ANS1        :STORE AC1 IN ANS1 THEN ANS4
005064 022767 040000 173710      CMP      #040000,ANS1     :DID 040000 GET STORED?
005072 001401                      BEQ      .+4             :BRANCH IF OK
005074 104004                      HLT+4          :ANS1 NOT EQUAL TO 040000

005076 022767 000000 173700      CMP      #000000,ANS2     :DID 000000 GET STORED?
005104 001401                      BEQ      .+4             :BRANCH IF OK
005106 104004                      HLT+4          :ANS2 NOT EQUAL TO 000000

005110 005767 173672      TST      ANS3            :DID ZERO GET STORED?
005114 001401                      BEQ      .+4             :BRANCH IF OK
005116 104004                      HLT+4          :ANS3 NOT EQUAL TO ZERO

005120 005767 173664      TST      ANS4            :DID ZERO GET STORED?
005124 001401                      BEQ      .+4             :BRANCH IF OK
005126 104004                      HLT+4          :ANS4 NOT EQUAL TO ZERO

```

```

:*****
:TEST 40:      TEST LDCFD (LOAD-CONVERT FLOATING TO DOUBLE)
:              LOAD  140000,000000 --> 140000,000000,000000,000000
:              FPS = 047610, SRC = M6-R7, AC = AC0
:*****

```

```

005130 104400          SCOPE
005132 000402          BR      TST40

005134 140000 000000   DAT40: 140000,000000

005140 170127 047617   TST40: LDFPS #047617      ;LOAD FLOATING POINT STAUS
005144 172404          LDD      AC4,   AC0      ;PRELOAD AC0 WITH GARBAGE
005146 177467 177762   FPI40: LDCFD  DAT40, AC0      ;LOAD-CONVERT 140000,000000 INTO AC0
005152 170200          STFPS  FPS          ;STORE FLOATING POINT STATUS
005154 022700 047610   CMP      #047610,FPS     ;CHECK FLOATING POINT STATUS
005160 001401          BEQ      .+4          ;BRANCH IF OK
005162 104000          HLT          ;FPS NOT EQUAL TO 047610

005164 174067 173612   STD      AC0,   ANS1     ;STORE AC0 IN ANS1 THRU ANS4
005170 022767 140000 173504  CMP      #140000,ANS1    ;DID 140000 GET STORED?
005176 001401          BEQ      .+4          ;BRANCH IF OK
005200 104004          HLT+4        ;ANS1 NOT EQUAL TO 140000

005202 022767 000000 173574  CMP      #000000,ANS2    ;DID 000000 GET STORED?
005210 001401          BEQ      .+4          ;BRANCH IF OK
005212 104004          HLT+4        ;ANS2 NOT EQUAL TO 000000

005214 005767 173566   TST      ANS3           ;DID ZERO GET STORED?
005220 001401          BEQ      .+4          ;BRANCH IF OK
005222 104004          HLT+4        ;ANS3 NOT EQUAL TO ZERO

005224 005767 173560   TST      ANS4           ;DID ZERO GET STORED?
005230 001401          BEQ      .+4          ;BRANCH IF OK
005232 104004          HLT+4        ;ANS4 NOT EQUAL TO ZERO

```

```

*****
:TEST 41:      TEST LDCFD (LOAD-CONVERT FLOATING TO DOUBLE)
:      LOAD 100000,000000 --> 125252,125252,122525,052525
:      FPS = 147614,   SRC = M6-R7,   AC = AC2
:      FEC = 14,     FEA = FPI41
*****

```

```

005234 104400          SCOPE
005236 000406          BR      TST41

005240 125252 125252 122525 SET41: 125252,125252,122525,052525
005246 052525
005250 100000 000000   DAT41: 100000,000000

005254 170127 047617   TST41: LDFPS #047617      ;LOAD FLOATING POINT STAUS
005260 172667 177754          LDD      SET41, AC2      ;LOAD 125252,125252,122525,052525 INTO AC2
005264 177667 177760   FPI41: LDCFD  DAT41, AC2      ;LOAD-CONVERT 100000,000000 INTO AC2
005270 170200          STFPS  FPS          ;STORE FLOATING POINT STATUS
005272 170367 173524   STST   FEC          ;STORE EXCEPTION CODES
005276 022700 147614   CMP      #147614,FPS     ;CHECK FLOATING POINT STATUS
005302 001401          BEQ      .+4          ;BRANCH IF OK
005304 104000          HLT          ;FPS NOT EQUAL TO 147614

005306 022767 000014 173506  CMP      #14,   FEC     ;CHECK FLOATING EXCEPTION CODE

```

F03

MAINDEC-11-DOCFPI-B
DOCFPI.P11

TEST OF LDCFD, LDCFD, STCFD, STCDF
TEST SECTION

MACY11 27(732) 17-SEP-76 10:46 PAGE 31

```

005314 001401          BEQ      .+4          :BRANCH IF OK
005316 104000          HLT                    :FEC NOT EQUAL TO 14

005320 022767 005264 173476  CMP      #FPI41, FEA      :CHECK FLOATING EXCEPTION ADDRESS
005326 001401          BEQ      .+4          :BRANCH IF OK
005330 104000          HLT                    :FEA NOT EQUAL TO FPI41

005332 174267 173444          STD      AC2, ANS1      :STORE AC2 IN ANS1 THRU ANS4
005336 022767 125252 173436  CMP      #125252, ANS1  :DID 125252 GET STORED?
005344 001401          BEQ      .+4          :BRANCH IF OK
005346 104004          HLT+4        :ANS1 NOT EQUAL TO 125252

005350 022767 125252 173426  CMP      #125252, ANS2  :DID 125252 GET STORED?
005356 001401          BEQ      .+4          :BRANCH IF OK
005360 104004          HLT+4        :ANS2 NOT EQUAL TO 125252

005362 022767 122525 173416  CMP      #122525, ANS3  :DID 122525 GET STORED?
005370 001401          BEQ      .+4          :BRANCH IF OK
005372 104004          HLT+4        :ANS3 NOT EQUAL TO 122525

005374 022767 052525 173406  CMP      #052525, ANS4  :DID 052525 GET STORED?
005402 001401          BEQ      .+4          :BRANCH IF OK
005404 104004          HLT+4        :ANS4 NOT EQUAL TO 052525

```

```

*****
:TEST 42:      TEST LDCFD (LOAD-CONVERT FLOATING TO DOUBLE)
:      LOAD    077777,177777 --> 077777,177777,000000,000000
:      FPS = 047600, SRC = M6-R7, AC = AC2
*****

```

```

005406 104400          SCOPE
005410 000402          BR      TST42

005412 077777 177777          DAT42: 077777,177777

005416 170127 047617          TST42: LDFPS  #047617      :LOAD FLOATING POINT STATUS
005422 172604          LDD      AC4, AC2      :PRELOAD AC2 WITH GARBAGE
005424 177667 177762          FPI42: LDCFD  DAT42, AC2  :LOAD-CONVERT 077777,177777 INTO AC2
005430 170200          STFPS   FPS           :STORE FLOATING POINT STATUS
005432 022700 047600          CMP      #047600, FPS  :CHECK FLOATING POINT STATUS
005436 001401          BEQ      .+4          :BRANCH IF OK
005440 104000          HLT                    :FPS NOT EQUAL TO 047600

005442 174267 173334          STD      AC2, ANS1      :STORE AC2 IN ANS1 THRU ANS4
005446 022767 077777 173326  CMP      #077777, ANS1  :DID 077777 GET STORED?
005454 001401          BEQ      .+4          :BRANCH IF OK
005456 104004          HLT+4        :ANS1 NOT EQUAL TO 077777

005460 022767 177777 173316  CMP      #177777, ANS2  :DID 177777 GET STORED?
005466 001401          BEQ      .+4          :BRANCH IF OK
005470 104004          HLT+4        :ANS2 NOT EQUAL TO 177777

005472 005767 173310          TST      ANS3          :DID ZERO GET STORED?
005476 001401          BEQ      .+4          :BRANCH IF OK
005500 104004          HLT+4        :ANS3 NOT EQUAL TO ZERO

```



```

005502 005767 173302      TST      ANS4      :DID ZERO GET STORED?
005506 001401           BEQ      .+4       :BRANCH IF OK
005510 104004           HLT+4           :ANS4 NOT EQUAL TO ZERO

```

```

:*****
:TEST 43:      TEST LDCFD (LOAD-CONVERT FLOATING TO DOUBLE)
:      LOAD    177777,177777 --> 177777,177777,000000,000000
:      FPS = 047610, SRC = M6-R7, AC = AC1
:*****

```

```

005512 104400           SCOPE
005514 000402           BR      TST43

005516 177777 177777   DAT43: 177777,177777

005522 170127 047617   TST43: LDFPS #047617      :LOAD FLOATING POINT STAU
005526 172504           LDD      AC4, AC1      :PRELOAD AC1 WITH GARBAGE
005530 177567 177762   FPI43: LDCFD  DAT43, AC1  :LOAD-CONVERT 177777,177777 INTO AC1
005534 170200           STFPS   FPS           :STORE FLOATING POINT STATUS
005536 022700 047610   CMP      #047610,FPS   :CHECK FLOATING POINT STATUS
005542 001401           BEQ      .+4          :BRANCH IF OK
005544 104000           HLT           :FPS NOT EQUAL TO 047610

```

```

005546 174167 173230   STD      AC1, ANS1     :STORE AC1 IN ANS1 THRU ANS4
005552 022767 177777 173222 CMP      #177777,ANS1  :DID 177777 GET STORED?
005560 001401           BEQ      .+4          :BRANCH IF OK
005562 104004           HLT+4           :ANS1 NOT EQUAL TO 177777

```

```

005564 022767 177777 173212 CMP      #177777,ANS2  :DID 177777 GET STORED?
005572 001401           BEQ      .+4          :BRANCH IF OK
005574 104004           HLT+4           :ANS2 NOT EQUAL TO 177777

```

```

005576 005767 173204   TST      ANS3      :DID ZERO GET STORED?
005602 001401           BEQ      .+4       :BRANCH IF OK
005604 104004           HLT+4           :ANS3 NOT EQUAL TO ZERO

```

```

005606 005767 173176   TST      ANS4      :DID ZERO GET STORED?
005612 001401           BEQ      .+4       :BRANCH IF OK
005614 104004           HLT+4           :ANS4 NOT EQUAL TO ZERO

```

```

:*****
:TEST 44:      TEST LDCFD (LOAD-CONVERT FLOATING TO DOUBLE)
:      LOAD    052525,052525 --> 052525,052525,000000,000000
:      FPS = 047600, SRC = M0-AC2, AC = AC0
:*****

```

```

005616 104400           SCOPE
005620 000402           BR      TST44

005622 052525 052525   DAT44: 052525,052525

005626 170127 047617   TST44: LDFPS #047617      :LOAD FLOATING POINT STAU

```

H03

MAINDEC-11-DCFP1-B
DCFP1.P11

TEST OF LDCFD, LDCFD, STCFD, STCDF
TEST SECTION

MACY11 27(732) 17-SEP-76 10:46 PAGE 33

```

005632 172467 177764      LDD      DAT44,  AC0      ;LOAD 052525,052525 INTO AC0
005636 177600             LDCFD    AC0,    AC2      ;LOAD-CONVERT AC0 INTO AC2
005640 170200             STFPS   FPS              ;STORE FLOATING POINT STATUS
005642 022700 047600     CMP      #047600,FPS     ;CHECK FLOATING POINT STATUS
005646 001401             BEQ      .+4              ;BRANCH IF OK
005650 104000             HLT                      ;FPS NOT EQUAL TO 047600

005652 174267 173124      STD      AC2,    ANS1     ;STORE AC2 IN ANS1 THRU ANS4
005656 022767 052525 173116  CMP      #052525,ANS1    ;DID 052525 GET STORED?
005664 001401             BEQ      .+4              ;BRANCH IF OK
005666 104004             HLT+4                    ;ANS1 NOT EQUAL TO 052525

005670 022767 052525 173106  CMP      #052525,ANS2    ;DID 052525 GET STORED?
005676 001401             BEQ      .+4              ;BRANCH IF OK
005700 104004             HLT+4                    ;ANS2 NOT EQUAL TO 052525

005702 005767 173100      TST      ANS3            ;DID ZERO GET STORED?
005706 001401             BEQ      .+4              ;BRANCH IF OK
005710 104004             HLT+4                    ;ANS3 NOT EQUAL TO ZERO

005712 005767 173072      TST      ANS4            ;DID ZERO GET STORED?
005716 001401             BEQ      .+4              ;BRANCH IF OK
005720 104004             HLT+4                    ;ANS4 NOT EQUAL TO ZERO

```

```

*****
:TEST 45:      TEST LDCFD (LOAD-CONVERT FLOATING TO DOUBLE)
:              LOAD 125252,125252 --> 125252,125252,000000,000000
:              FPS = 047610, SRC = M1-R2, AC = AC0
*****

```

```

005722 104400             SCOPE
005724 000402             BR      TST45

005726 125252 125252      DAT45: 125252,125252

005732 170127 047617      TST45: LDFPS   #047617      ;LOAD FLOATING POINT STAUS
005736 012702 005726      MOV      #DAT45, R2
005742 177412             LDCFD    (R2),  AC0      ;LOAD-CONVERT 125252,125252 INTO AC0
005744 170200             STFPS   FPS              ;STORE FLOATING POINT STATUS
005746 022700 047610     CMP      #047610,FPS     ;CHECK FLOATING POINT STATUS
005752 001401             BEQ      .+4              ;BRANCH IF OK
005754 104000             HLT                      ;FPS NOT EQUAL TO 047610

005756 174067 173020      STD      AC0,    ANS1     ;STORE AC0 IN ANS1 THRU ANS4
005762 022702 005726      CMP      #DAT45, R2      ;CHECK R2
005766 000401             BR      .+4              ;BRANCH IF OK
005770 104000             HLT                      ;R2 NOT EQUAL TO #DAT45

005772 022767 125252 173002  CMP      #125252,ANS1    ;DID 125252 GET STORED?
006000 001401             BEQ      .+4              ;BRANCH IF OK
006002 104004             HLT+4                    ;ANS1 NOT EQUAL TO 125252

006004 022767 125252 172772  CMP      #125252,ANS2    ;DID 125252 GET STORED?
006012 001401             BEQ      .+4              ;BRANCH IF OK
006014 104004             HLT+4                    ;ANS2 NOT EQUAL TO 125252

```

```

006016 005767 172764      TST      ANS3      ;DID ZERO GET STORED?
006022 001401      BEQ      .+4       ;BRANCH IF OK
006024 104004      HLT+4     ;ANS3 NOT EQUAL TO ZERO

006026 005767 172756      TST      ANS4      ;DID ZERO GET STORED?
006032 001401      BEQ      .+4       ;BRANCH IF OK
006034 104004      HLT+4     ;ANS4 NOT EQUAL TO ZERO

```

```

*****
:TEST 46:      TEST LDCFD (LOAD-CONVERT FLOATING TO DOUBLE)
:              LOAD      040252,125252 --> 040252,125252,000000,000000
:              FPS = 047600, SRC = M2-R6, AC = AC2
:              *****

```

```

006036 104400      SCOPE
006040 000401      BR      TST46

006042 000000      SAV46: 0

006044 170127 047617      TST46: LDFPS    #047617      ;LOAD FLOATING POINT STATUS
006050 010667 177766      MOV     %6, SAV46      ;SAVE SP
006054 012746 125252      MOV     #125252,-(%6)  ;PUSH 125252 ON STACK
006060 012746 040252      MOV     #040252,-(%6)  ;PUSH 040252 ON STACK
006064 177626      LDCFD  (%6)+, AC2      ;LOAD-CONVERT 040252,125252 INTO AC2
006066 170200      STFPS  FPS             ;STORE FLOATING POINT STATUS
006070 020667 177746      CMP     %6, SAV46      ;CHECK SP
006074 001405      BEQ     TSA46          ;BRANCH IF OK
006076 010667 172700      MOV     %6, ANS1       ;SAVE WRONG %6 FOR TYPING
006102 016706 177734      MOV     SAV46, %6      ;RESTORE SP
006106 104001      HLT+1   ;STACK POINTER WRONG

006110 020027 047600      TSA46: CMP     FPS, #047600 ;CHECK FLOATING POINT STATUS
006114 001401      BEQ     .+4           ;BRANCH IF OK
006116 104000      HLT     ;FLOATING POINT STATUS NOT 047600

006120 174267 172656      STD     AC2, ANS1      ;STORE AC2 IN ANS1 THRU ANS4
006124 022767 040252 172650      CMP     #040252,ANS1  ;DID 040252 GET STORED?
006132 001401      BEQ     .+4           ;BRANCH IF OK
006134 104004      HLT+4   ;ANS1 NOT EQUAL TO 040252

006136 022767 125252 172640      CMP     #125252,ANS2  ;DID 125252 GET STORED?
006144 001401      BEQ     .+4           ;BRANCH IF OK
006146 104004      HLT+4   ;ANS2 NOT EQUAL TO 125252

006150 005767 172632      TST     ANS3          ;DID ZERO GET STORED?
006154 001401      BEQ     .+4           ;BRANCH IF OK
006156 104004      HLT+4   ;ANS3 NOT EQUAL TO ZERO

006160 005767 172624      TST     ANS4          ;DID ZERO GET STORED?
006164 001401      BEQ     .+4           ;BRANCH IF OK
006166 104004      HLT+4   ;ANS4 NOT EQUAL TO ZERO

```

```

*****

```

:TEST 47: TEST LDCFD (LOAD-CONVERT FLOATING TO DOUBLE)
: LOAD 140125,052525 --> 140125,052525,000000,000000
: FPS = 047610, SRC = M3-R2, AC = AC1
:*****

006170 104400
006172 000403

SCOPE
BR TST47

006174 140125 052525
006200 006174

DAT47: 140125,052525
ADR47: DAT47

006202 170127 047617
006206 012702 006200
006212 177532
006214 170200
006216 022700 047610
006222 001401
006224 104000

TST47: LDFPS #047617 ;LOAD FLOATING POINT STAUS
MOV #ADR47, R2 ;LOAD R2
LDCFD @(R2)+, AC1 ;LOAD-CONVERT 140125,052525 INTO AC1
STFPS FPS ;STORE FLOATING POINT STATUS
CMP #047610,FPS ;CHECK FLOATING POINT STATUS
BEQ .+4 ;BRANCH IF OK
HLT ;FPS NOT EQUAL TO 047610

006226 174167 172550
006232 022702 006202
006236 001401
006240 104000

STD AC1, ANS1 ;STORE AC1 IN ANS1 THRU ANS4
CMP #TST47, R2 ;CHECK R2
BEQ .+4 ;BRANCH IF OK
HLT ;R2 NOT EQUAL TO #TST47

006242 022767 140125 172532
006250 001401
006252 104004

CMP #140125,ANS1 ;DID 140125 GET STORED?
BEQ .+4 ;BRANCH IF OK
HLT+4 ;ANS1 NOT EQUAL TO 140125

006254 022767 052525 172522
006262 001401
006264 104004

CMP #052525,ANS2 ;DID 052525 GET STORED?
BEQ .+4 ;BRANCH IF OK
HLT+4 ;ANS2 NOT EQUAL TO 052525

006266 005767 172514
006272 001401
006274 104004

TST ANS3 ;DID ZERO GET STORED?
BEQ .+4 ;BRANCH IF OK
HLT+4 ;ANS3 NOT EQUAL TO ZERO

006276 005767 172506
006302 001401
006304 104004

TST ANS4 ;DID ZERO GET STORED?
BEQ .+4 ;BRANCH IF OK
HLT+4 ;ANS4 NOT EQUAL TO ZERO

:*****
:TEST 50: TEST LDCFD (LOAD-CONVERT FLOATING TO DOUBLE)
: LOAD 040125,052525 --> 040125,052525,000000,000000
: FPS = 047600, SRC = M4-R4, AC = AC2
:*****

006306 104400
006310 000402

SCOPE
BR TST50

006312 040125 052525

DAT50: 040125,052525

006316 170127 047617
006322 012704 006316
006326 177644
006330 170200

TST50: LDFPS #047617 ;LOAD FLOATING POINT STAUS
MOV #TST50, R4 ;LOAD R4
LDCFD -(R4), AC2 ;LOAD-CONVERT 040125,052525 INTO AC2
STFPS FPS ;STORE FLOATING POINT STATUS

```

006332 022700 047600      CMP      #047600,FPS      ;CHECK FLOATING POINT STATUS
006336 001401      BEQ      .+4          ;BRANCH IF OK
006340 104000      HLT                      ;FPS NOT EQUAL TO 047600

006342 174267 172434      STD      AC2, ANS1     ;STORE AC2 IN ANS1 THRU ANS4
006346 022704 006312      CMP      #DAT50, R4    ;CHECK R4
006352 001401      BEQ      .+4          ;BRANCH IF OK
006354 104000      HLT                      ;R4 NOT EQUAL TO #DAT50

006356 022767 040125 172416  CMP      #040125,ANS1  ;DID 040125 GET STORED?
006364 001401      BEQ      .+4          ;BRANCH IF OK
006366 104004      HLT+4                ;ANS1 NOT EQUAL TO 040125

006370 022767 052525 172406  CMP      #052525,ANS2  ;DID 052525 GET STORED?
006376 001401      BEQ      .+4          ;BRANCH IF OK
006400 104004      HLT+4                ;ANS2 NOT EQUAL TO 052525

006402 005767 172400      TST      ANS3          ;DID ZERO GET STORED?
006406 001401      BEQ      .+4          ;BRANCH IF OK
006410 104004      HLT+4                ;ANS3 NOT EQUAL TO ZERO

006412 005767 172372      TST      ANS4          ;DID ZERO GET STORED?
006416 001401      BEQ      .+4          ;BRANCH IF OK
006420 104004      HLT+4                ;ANS4 NOT EQUAL TO ZERO

```

```

*****
:TEST 51:      TEST LDCFD (LOAD-CONVERT FLOATING TO DOUBLE)
:              LOAD 140252,125252 --> 140252,125252,000000,000000
:              FPS = 047610, SRC = M5-R3, AC = ACO
*****

```

```

006422 104400      SCOPE
006424 000403      BR      TST51

006426 140252 125252  DAT51: 140252,125252
006432 006426  ADR51: DAT51

006434 170127 047617  TST51: LDFPS #047617      ;LOAD FLOATING POINT STAUS
006440 012703 006434      MOV      #TST51, R3    ;LOAD R3
006444 177453      LDCFD   @-(R3), ACO   ;LOAD-CONVERT 140252,125252 INTO ACO
006446 170200      STFPS  FPS           ;STORE FLOATING POINT STATUS
006450 022700 047610  CMP      #047610,FPS  ;CHECK FLOATING POINT STATUS
006454 001401      BEQ      .+4          ;BRANCH IF OK
006456 104000      HLT                      ;FPS NOT EQUAL TO 047610

006460 174067 172316  STD      ACO, ANS1     ;STORE ACO IN ANS1 THRU ANS4
006464 022703 006432  CMP      #ADR51, R3    ;CHECK R3
006470 001401      BEQ      .+4          ;BRANCH IF OK
006472 104000      HLT                      ;R3 NOT EQUAL TO #ADR51

006474 022767 140252 172300  CMP      #140252,ANS1  ;DID 140252 GET STORED?
006502 001401      BEQ      .+4          ;BRANCH IF OK
006504 104004      HLT+4                ;ANS1 NOT EQUAL TO 140252

006506 022767 125252 172270  CMP      #125252,ANS2  ;DID 125252 GET STORED?

```

MAINDEC-11-DCFPI-B
DCFPI.P11

TEST OF LDCFD, LDCFD, STCFD, STCDF
TEST SECTION

```

006514 001401      BEQ      .+4      ;BRANCH IF OK
006516 104004      HLT+4          ;ANS2 NOT EQUAL TO 125252

006520 005767 172262 TST      ANS3      ;DID ZERO GET STORED?
006524 001401      BEQ      .+4      ;BRANCH IF OK
006526 104004      HLT+4          ;ANS3 NOT EQUAL TO ZERO

006530 005767 172254 TST      ANS4      ;DID ZERO GET STORED?
006534 001401      BEQ      .+4      ;BRANCH IF OK
006536 104004      HLT+4          ;ANS4 NOT EQUAL TO ZERO

```

```

*****
:TEST 52:      TEST LDCFD (LOAD-CONVERT FLOATING TO DOUBLE)
:      LOAD      052525,052525 --> 052525,052525,000000,000000
:      FPS = 047600, SRC = M6-R5, AC = AC2
*****

```

```

006540 104400      SCOPE
006542 000402      BR      TST52

006544 052525 052525 DAT52: 052525,052525

006550 170127 047617 TST52: LDFPS #047617 ;LOAD FLOATING POINT STATUS
006554 012705 006531 MOV      #DAT52-13,R5 ;LOAD R5
006560 177665 000013 LDCFD   13(R5), AC2 ;LOAD-CONVERT 052525,052525 INTO AC2
006564 170200 STFPS   FPS ;STORE FLOATING POINT STATUS
006566 022700 047600 CMP      #047600,FPS ;CHECK FLOATING POINT STATUS
006572 001401 BEQ      .+4 ;BRANCH IF OK
006574 104000 HLT          ;FPS NOT EQUAL TO 047600

006576 174267 172200 STD      AC2, ANS1 ;STORE AC2 IN ANS1 THRU ANS4
006602 022705 006531 CMP      #DAT52-13,R5 ;CHECK R5
006606 001401 BEQ      .+4 ;BRANCH IF OK
006610 104000 HLT          ;R5 NOT EQUAL TO #DAT52-13

006612 022767 052525 172162 CMP      #052525,ANS1 ;DID 052525 GET STORED?
006620 001401 BEQ      .+4 ;BRANCH IF OK
006622 104004 HLT+4          ;ANS1 NOT EQUAL TO 052525

006624 022767 052525 172152 CMP      #052525,ANS2 ;DID 052525 GET STORED?
006632 001401 BEQ      .+4 ;BRANCH IF OK
006634 104004 HLT+4          ;ANS2 NOT EQUAL TO 052525

006636 005767 172144 TST      ANS3      ;DID ZERO GET STORED?
006642 001401 BEQ      .+4 ;BRANCH IF OK
006644 104004 HLT+4          ;ANS3 NOT EQUAL TO ZERO

006646 005767 172136 TST      ANS4      ;DID ZERO GET STORED?
006652 001401 BEQ      .+4 ;BRANCH IF OK
006654 104004 HLT+4          ;ANS4 NOT EQUAL TO ZERO

```

```

*****
:TEST 53:      TEST LDCFD (LOAD-CONVERT FLOATING TO DOUBLE)
:      LOAD      125252,125252 --> 125252,125252,000000,000000

```

: FPS = 047610, SRC = M7-R2, AC = AC0
:*****

006656 104400
006660 000403

SCOPE
BR TST53

006662 125252 125252

DAT53: 125252,125252

006666 006662
006670 170127 047617
006674 012702 006670
006700 177472 177776
006704 170200
006706 022700 047610
006712 001401
006714 104000

ADR53: DAT53
TST53: LDFPS #047617 ;LOAD FLOATING POINT STAUS
MOV #ADR53+2,R2 ;LOAD R2
LDCFD 2-2(R2),AC0 ;LOAD-CONVERT 125252,125252 INTO AC0
STFPS FPS ;STORE FLOATING POINT STATUS
CMP #047610,FPS ;CHECK FLOATING POINT STATUS
BEQ .+4 ;BRANCH IF OK
HLT ;FPS NOT EQUAL TO 047610

006716 174067 172060
006722 022702 006670
006726 001401
006730 104000

STD AC0, ANS1 ;STORE AC0 IN ANS1 THRU ANS4
CMP #ADR53+2,R2 ;CHECK R2
BEQ .+4 ;BRANCH IF OK
HLT ;R2 NOT EQUAL TO #ADR53+2

006732 022767 125252 172042
006740 001401
006742 104004

CMP #125252,ANS1 ;DID 125252 GET STORED?
BEQ .+4 ;BRANCH IF OK
HLT+4 ;ANS1 NOT EQUAL TO 125252

006744 022767 125252 172032
006752 001401
006754 104004

CMP #125252,ANS2 ;DID 125252 GET STORED?
BEQ .+4 ;BRANCH IF OK
HLT+4 ;ANS2 NOT EQUAL TO 125252

006756 005767 172024
006762 001401
006764 104004

TST ANS3 ;DID ZERO GET STORED?
BEQ .+4 ;BRANCH IF OK
HLT+4 ;ANS3 NOT EQUAL TO ZERO

006766 005767 172016
006772 001401
006774 104004

TST ANS4 ;DID ZERO GET STORED?
BEQ .+4 ;BRANCH IF OK
HLT+4 ;ANS4 NOT EQUAL TO ZERO

:*****
:TEST 54: TEST LDCFD (LOAD-CONVERT FLOATING TO DOUBLE)
:-47615 GETS ASSEMBLED AS 144072
:LOAD 144072 --> 144072,000000,000000
:FPS = 047610, SRC = M2-R7, AC = AC1
:*****

006776 104400

SCOPE

007000 170127 047617
007004 177527 144072
007010 170200
007012 022700 047610
007016 001401
007020 104000

TST54: LDFPS #047617 ;LOAD FLOATING POINT STAUS
LDCFD #-47615,AC1 ;LOAD-CONVERT -47615 INTO AC1
STFPS FPS ;STORE FLOATING POINT STATUS
CMP #047610,FPS ;CHECK FLOATING POINT STATUS
BEQ .+4 ;BRANCH IF OK
HLT ;FPS NOT EQUAL TO 047610

007022 174167 171754

STD AC1, ANS1 ;STORE AC1 IN ANS1 THRU ANS4

MAINDEC-11-DCFPI-B
DCFPI.P11

TEST OF LDCFD, LDCFD, STCFD, STCFD
TEST SECTION

007026	022767	144072	171746	CMP	#144072,ANS1	;DID 144072 GET STORED?
007034	001401			BEQ	+.4	;BRANCH IF OK
007036	104004			HLT+4		;ANS1 NOT EQUAL TO 144072
007040	005767	171740		TST	ANS2	;DID ZERO GET STORED?
007044	001401			BEQ	+.4	;BRANCH IF OK
007046	104004			HLT+4		;ANS2 NOT EQUAL TO ZERO
007050	005767	171732		TST	ANS3	;DID ZERO GET STORED?
007054	001401			BEQ	+.4	;BRANCH IF OK
007056	104004			HLT+4		;ANS3 NOT EQUAL TO ZERO
007060	005767	171724		TST	ANS4	;DID ZERO GET STORED?
007064	001401			BEQ	+.4	;BRANCH IF OK
007066	104004			HLT+4		;ANS4 NOT EQUAL TO ZERO

```

*****
:TEST 55:      TEST LDCFD (LOAD-CONVERT FLOATING TO DOUBLE)
:              LOAD  040125,052525 --> 040125,052525,000000,000000
:              FPS = 047600, SRC = M3-R7, AC = AC3
*****

```

007070	104400			SCOPE		
007072	000402			BR	TST55	
007074	040125	052525		DAT55:	040125,052525	
007100	170127	047617		TST55:	LDFPS #047617	;LOAD FLOATING POINT STATUS
007104	177737	007074			LDCFD @DAT55,AC3	;LOAD-CONVERT 040125,052525 INTO AC3
007110	170200				STFPS FPS	;STORE FLOATING POINT STATUS
007112	022700	047600			CMP #047600,FPS	;CHECK FLOATING POINT STATUS
007116	001401				BEQ .+4	;BRANCH IF OK
007120	104000				HLT	;FPS NOT EQUAL TO 047600
007122	174367	171654		STD	AC3, ANS1	;STORE AC3 IN ANS1 THRU ANS4
007126	022767	040125	171646	CMP	#040125,ANS1	;DID 040125 GET STORED?
007134	001401			BEQ	+.4	;BRANCH IF OK
007136	104004			HLT+4		;ANS1 NOT EQUAL TO 040125
007140	022767	052525	171636	CMP	#052525,ANS2	;DID 052525 GET STORED?
007146	001401			BEQ	+.4	;BRANCH IF OK
007150	104004			HLT+4		;ANS2 NOT EQUAL TO 052525
007152	005767	171630		TST	ANS3	;DID ZERO GET STORED?
007156	001401			BEQ	+.4	;BRANCH IF OK
007160	104004			HLT+4		;ANS3 NOT EQUAL TO ZERO
007162	005767	171622		TST	ANS4	;DID ZERO GET STORED?
007166	001401			BEQ	+.4	;BRANCH IF OK
007170	104004			HLT+4		;ANS4 NOT EQUAL TO ZERO

```

*****
:TEST 56:      TEST LDCFD (LOAD-CONVERT FLOATING TO DOUBLE)
:              LOAD  140252,125252 --> 140252,125252,000000,000000
*****

```

: FPS = 047610, SRC = M7-R7, AC = ACC

```

007172 104400          SCOPE
007174 000403          BR      TST56

007176 140252 125252   DAT56: 140252,125252
007202 007176          ADR56: DAT56

007204 170127 047617   TST56: LDFPS  #047617      ;LOAD FLOATING POINT STAU
007210 177477 177766   LDCFD  @ADR56, ACC      ;LOAD-CONVERT 140252,125252 INTO ACC
007214 170200          STFPS  FPS              ;STORE FLOATING POINT STATUS
007216 022700 047610   CMP    #047610,FPS      ;CHECK FLOATING POINT STATUS
007222 001401          BEQ    .+4              ;BRANCH IF OK
007224 104000          HLT                    ;FPS NOT EQUAL TO 047610

007226 174067 171550   STD    ACC, ANS1        ;STORE ACC IN ANS1 THRU ANS4
007232 022767 140252 171542  CMP    #140252,ANS1     ;DID 140252 GET STORED?
007240 001401          BEQ    .+4              ;BRANCH IF OK
007242 104004          HLT+4                    ;ANS1 NOT EQUAL TO 140252

007244 022767 125252 171532  CMP    #125252,ANS2     ;DID 125252 GET STORED?
007252 001401          BEQ    .+4              ;BRANCH IF OK
007254 104004          HLT+4                    ;ANS2 NOT EQUAL TO 125252

007256 005767 171524   TST    ANS3             ;DID ZERO GET STORED?
007262 001401          BEQ    .+4              ;BRANCH IF OK
007264 104004          HLT+4                    ;ANS3 NOT EQUAL TO ZERO

007266 005767 171516   TST    ANS4             ;DID ZERO GET STORED?
007272 001401          BEQ    .+4              ;BRANCH IF OK
007274 104004          HLT+4                    ;ANS4 NOT EQUAL TO ZERO

```

: TEST S7: TEST STCFD (STORE-CONVERT FLOATING TO DOUBLE)
: STORE 000000,000000 --> 000000,000000,000000,000000
: FPS = 047404, AC = AC1, DST = M6-R7

```

007276 104400          SCOPE
007300 000402          BR      TST57

007302 000000 000000   DAT57: 000000,000000

007306 170127 047417   TST57: LDFPS  #047417      ;LOAD FLOATING POINT STAU
007312 172567 177764   LDF    DAT57, AC1      ;LOAD 000000,000000 INTO AC1
007316 176167 171460   FP157: STCFD  AC1, ANS1     ;STORE-CONVERT AC1 IN ANS1 THRU ANS4
007322 170200          STFPS  FPS              ;STORE FLOATING POINT STATUS
007324 022700 047404   CMP    #047404,FPS      ;CHECK FLOATING POINT STATUS
007330 001401          BEQ    .+4              ;BRANCH IF OK
007332 104000          HLT                    ;FPS NOT EQUAL TO 047404

007334 022767 000000 171440  CMP    #000000,ANS1     ;DID 000000 GET STORED?
007342 001401          BEQ    .+4              ;BRANCH IF OK
007344 104004          HLT+4                    ;ANS1 NOT EQUAL TO 000000

```

```

007346 022767 000000 171430      CMP      #000000,ANS2      :DID 000000 GET STORED?
007354 001401                      BEQ      .+4              :BRANCH IF OK
007356 104004                      HLT+4                    :ANS2 NOT EQUAL TO 000000

007360 005767 171422      TST      ANS3              :DID ZERO GET STORED?
007364 001401                      BEQ      .+4              :BRANCH IF OK
007366 104004                      HLT+4                    :ANS3 NOT EQUAL TO ZERO

007370 005767 171414      TST      ANS4              :DID ZERO GET STORED?
007374 001401                      BEQ      .+4              :BRANCH IF OK
007376 104004                      HLT+4                    :ANS4 NOT EQUAL TO ZERO

```

```

:*****
:TEST 60:      TEST STCFD (STORE-CONVERT FLOATING TO DOUBLE)
:      STORE  052525,052525 --> 052525,052525,000000,000000
:      FPS = 047400, AC = AC2, DST = M6-R7
:*****

```

```

007400 104400      SCOPE
007402 000402      BR      TST60

007404 052525 052525      DAT60: 052525,052525

007410 170127 047417      TST60: LDFPS #047417      :LOAD FLOATING POINT STATUS
007414 172667 177764      LDF      DAT60, AC2      :LOAD 052525,052525 INTO AC2
007420 176267 171356      FPI60: STCFD AC2, ANS1      :STORE-CONVERT AC2 IN ANS1 THRU ANS4
007424 170200      STFPS      FPS          :STORE FLOATING POINT STATUS
007426 022700 047400      CMP      #047400,FPS      :CHECK FLOATING POINT STATUS
007432 001401                      BEQ      .+4              :BRANCH IF OK
007434 104000                      HLT                    :FPS NOT EQUAL TO 047400

007436 022767 052525 171336      CMP      #052525,ANS1      :DID 052525 GET STORED?
007444 001401                      BEQ      .+4              :BRANCH IF OK
007446 104004                      HLT+4                    :ANS1 NOT EQUAL TO 052525

007450 022767 052525 171326      CMP      #052525,ANS2      :DID 052525 GET STORED?
007456 001401                      BEQ      .+4              :BRANCH IF OK
007460 104004                      HLT+4                    :ANS2 NOT EQUAL TO 052525

007462 005767 171320      TST      ANS3              :DID ZERO GET STORED?
007466 001401                      BEQ      .+4              :BRANCH IF OK
007470 104004                      HLT+4                    :ANS3 NOT EQUAL TO ZERO

007472 005767 171312      TST      ANS4              :DID ZERO GET STORED?
007476 001401                      BEQ      .+4              :BRANCH IF OK
007500 104004                      HLT+4                    :ANS4 NOT EQUAL TO ZERO

```

```

:*****
:TEST 61:      TEST STCFD (STORE-CONVERT FLOATING TO DOUBLE)
:      STORE  125252,125252 --> 125252,125252,000000,000000
:      FPS = 047410, AC = AC3, DST = M6-R7
:*****

```

```

007502 104400          SCOPE
007504 000402          BR      TST61

007506 125252 125252  DAT61: 125252,125252

007512 170127 047417  TST61: LDFPS #047417      :LOAD FLOATING POINT STAU
007516 172767 177764  LDF   DAT61, AC3      :LOAD 125252,125252 INTO AC3
007522 176367 171254  FPI61: STCFD AC3, ANS1  :STORE-CONVERT AC3 IN ANS1 THRU ANS4
007526 170200          STFPS FPS              :STORE FLOATING POINT STATUS
007530 022700 047410  CMP   #047410,FPS     :CHECK FLOATING POINT STATUS
007534 001401          BEQ   .+4              :BRANCH IF OK
007536 104000          HLT                    :FPS NOT EQUAL TO 047410

007540 022767 125252 171234  CMP   #125252,ANS1    :DID 125252 GET STORED?
007546 001401          BEQ   .+4              :BRANCH IF OK
007550 104004          HLT+4            :ANS1 NOT EQUAL TO 125252

007552 022767 125252 171224  CMP   #125252,ANS2    :DID 125252 GET STORED?
007560 001401          BEQ   .+4              :BRANCH IF OK
007562 104004          HLT+4            :ANS2 NOT EQUAL TO 125252

007564 005767 171216          TST   ANS3              :DID ZERO GET STORED?
007570 001401          BEQ   .+4              :BRANCH IF OK
007572 104004          HLT+4            :ANS3 NOT EQUAL TO ZERO

007574 005767 171210          TST   ANS4              :DID ZERO GET STORED?
007600 001401          BEQ   .+4              :BRANCH IF OK
007602 104004          HLT+4            :ANS4 NOT EQUAL TO ZERO

```

```

:*****
:TEST 62: TEST STCFD (STORE-CONVERT FLOATING TO DOUBLE)
:STORE 000200,000000 --> 000200,000000,000000,000000
:FPS = 047400, AC = AC1, DST = M6-R7
:*****

```

```

007604 104400          SCOPE
007606 000402          BR      TST62

007610 000200 000000  DAT62: 000200,000000

007614 170127 047417  TST62: LDFPS #047417      :LOAD FLOATING POINT STAU
007620 172567 177764  LDF   DAT62, AC1      :LOAD 000200,000000 INTO AC1
007624 176167 171152  FPI62: STCFD AC1, ANS1  :STORE-CONVERT AC1 IN ANS1 THRU ANS4
007630 170200          STFPS FPS              :STORE FLOATING POINT STATUS
007632 022700 047400  CMP   #047400,FPS     :CHECK FLOATING POINT STATUS
007636 001401          BEQ   .+4              :BRANCH IF OK
007640 104000          HLT                    :FPS NOT EQUAL TO 047400

007642 022767 000200 171132  CMP   #000200,ANS1    :DID 000200 GET STORED?
007650 001401          BEQ   .+4              :BRANCH IF OK
007652 104004          HLT+4            :ANS1 NOT EQUAL TO 000200

007654 022767 000000 171122  CMP   #000000,ANS2    :DID 000000 GET STORED?
007662 001401          BEQ   .+4              :BRANCH IF OK
007664 104004          HLT+4            :ANS2 NOT EQUAL TO 000000

```

E04

MAINDEC-11-DCFPI-8
DCFPI.P11

TEST OF LDCDF, LDCFD, STCFD, STCDF
TEST SECTION

MACY11 27(732) 17-SEP-76 10:46 PAGE 43

```

007666 005767 171114      TST      ANS3      ;DID ZERO GET STORED?
007672 001401              BEQ      .+4        ;BRANCH IF OK
007674 104004              HLT+4      ;ANS3 NOT EQUAL TO ZERO

007676 005767 171106      TST      ANS4      ;DID ZERO GET STORED?
007702 001401              BEQ      .+4        ;BRANCH IF OK
007704 104004              HLT+4      ;ANS4 NOT EQUAL TO ZERO

```

```

:*****
:TEST 63:      TEST STCFD (STORE-CONVERT FLOATING TO DOUBLE)
:              STORE 100200,000000 --> 100200,000000,000000,000000
:              FPS = 047410, AC = AC3, DST = M6-R7
:*****

```

```

007706 104400      SCOPE
007710 000402      BR      TST63

007712 100200 000000      DAT63: 100200,000000

007716 170127 047417      TST63: LDFPS #047417      ;LOAD FLOATING POINT STATUS
007722 172767 177764      LDF      DAT63, AC3      ;LOAD 100200,000000 INTO AC3
007726 176367 171050      FPI63: STCFD AC3, ANS1      ;STORE-CONVERT AC3 IN ANS1 THRU ANS4
007732 170200      STFPS      FPS          ;STORE FLOATING POINT STATUS
007734 022700 047410      CMP      #047410,FPS      ;CHECK FLOATING POINT STATUS
007740 001401      BEQ      .+4          ;BRANCH IF OK
007742 104000      HLT          ;FPS NOT EQUAL TO 047410

007744 022767 100200 171030      CMP      #100200,ANS1      ;DID 100200 GET STORED?
007752 001401      BEQ      .+4          ;BRANCH IF OK
007754 104004      HLT+4      ;ANS1 NOT EQUAL TO 100200

007756 022767 000000 171020      CMP      #000000,ANS2      ;DID 000000 GET STORED?
007764 001401      BEQ      .+4          ;BRANCH IF OK
007766 104004      HLT+4      ;ANS2 NOT EQUAL TO 000000

007770 005767 171012      TST      ANS3      ;DID ZERO GET STORED?
007774 001401      BEQ      .+4        ;BRANCH IF OK
007776 104004      HLT+4      ;ANS3 NOT EQUAL TO ZERO

010000 005767 171004      TST      ANS4      ;DID ZERO GET STORED?
010004 001401      BEQ      .+4        ;BRANCH IF OK
010006 104004      HLT+4      ;ANS4 NOT EQUAL TO ZERO

```

```

:*****
:TEST 64:      TEST STCFD (STORE-CONVERT FLOATING TO DOUBLE)
:              STORE 000177,177777 --> 000000,000000,000000,000000
:              FPS = 047404, AC = AC0, DST = M6-R7
:*****

```

```

010010 104400      SCOPE
010012 000402      BR      TST64

010014 000177 177777      DAT64: 000177,177777

```

010020	170127	000000	TST64:	LDFPS	#0		:CLEAR FLOATING POINT STATUS
010024	172467	177764		LDF	DAT64,	AC0	:LOAD 000177,177777 INTO AC0
010030	170127	047417		LDFPS	#047417		:LOAD FLOATING POINT STATUS
010034	176067	170742	FPI64:	STCFD	AC0,	ANS1	:STORE-CONVERT AC0 IN ANS1 THRU ANS4
010040	170200			STFPS	FPS		:STORE FLOATING POINT STATUS
010042	022700	047404		CMP	#047404,FPS		:CHECK FLOATING POINT STATUS
010046	001401			BEQ	+.4		:BRANCH IF OK
010050	104000			HLT			:FPS NOT EQUAL TO 047404
010052	005767	170724		TST	ANS1		:CHECK ANS1
010056	001401			BEQ	+.4		:BRANCH IF OK
010060	104004			HLT+4			:ANS1 NOT EQUAL TO 000177
010062	005767	170716		TST	ANS2		:CHECK ANS2
010066	001401			BEQ	+.4		:BRANCH IF OK
010070	104004			HLT+4			:ANS2 NOT EQUAL TO 177777
010072	005767	170710		TST	ANS3		:DID ZERO GET STORED?
010076	001401			BEQ	+.4		:BRANCH IF OK
010100	104004			HLT+4			:ANS3 NOT EQUAL TO ZERO
010102	005767	170702		TST	ANS4		:DID ZERO GET STORED?
010106	001401			BEQ	+.4		:BRANCH IF OK
010110	104004			HLT+4			:ANS4 NOT EQUAL TO ZERO

```

*****
:TEST 65: TEST STCFD (STORE-CONVERT FLOATING TO DOUBLE)
:STORE 100165,125252 --> 000000,000000,000000,000000
:FPS = 047404, AC = AC2, DST = M6-R7
*****

```

010112	104400			SCOPE			
010114	000402			BR	TST65		
010116	100165	125252	DAT65:	100165,125252			
010122	170127	000000	TST65:	LDFPS	#0		:CLEAR FLOATING POINT STATUS
010126	172667	177764		LDF	DAT65,	AC2	:LOAD 100165,125252 INTO AC2
010132	170127	047417		LDFPS	#047417		:LOAD FLOATING POINT STATUS
010136	176267	170640	FPI65:	STCFD	AC2,	ANS1	:STORE-CONVERT AC2 IN ANS1 THRU ANS4
010142	170200			STFPS	FPS		:STORE FLOATING POINT STATUS
010144	022700	047404		CMP	#047404,FPS		:CHECK FLOATING POINT STATUS
010150	001401			BEQ	+.4		:BRANCH IF OK
010152	104000			HLT			:FPS NOT EQUAL TO 047404
010154	005767	170622		TST	ANS1		:CHECK ANS1
010160	001401			BEQ	+.4		:BRANCH IF OK
010162	104004			HLT+4			:ANS1 NOT EQUAL TO 100165
010164	005767	170614		TST	ANS2		:CHECK ANS2
010170	001401			BEQ	+.4		:BRANCH IF OK
010172	104004			HLT+4			:ANS2 NOT EQUAL TO 125252
010174	005767	170606		TST	ANS3		:DID ZERO GET STORED?

```

010200 001401      BEQ      .+4      :BRANCH IF OK
010202 104004      HLT+4      :ANS3 NOT EQUAL TO ZERO

010204 005767 170600 TST      ANS4      :DID ZERO GET STORED?
010210 001401      BEQ      .+4      :BRANCH IF OK
010212 104004      HLT+4      :ANS4 NOT EQUAL TO ZERO

```

```

:*****
:TEST 66:      TEST STCFD (STORE-CONVERT FLOATING TO DOUBLE)
:      STORE  040000,000000 --> 040000,000000,000000,000000
:      FPS = 047400, AC = AC2,      DST = M6-R7
:*****

```

```

010214 104400      SCOPE
010216 000402      BR      TST66

010220 040000 000000 DAT66: 040000,000000

010224 170127 047417 TST66: LDFPS #047417 :LOAD FLOATING POINT STATUS
010230 172667 177764 LDF  DAT66, AC2 :LOAD 040000,000000 INTO AC2
010234 176267 170542 FPI66: STCFD AC2, ANS1 :STORE-CONVERT AC2 IN ANS1 THRU ANS4
010240 170200 STFPS FPS :STORE FLOATING POINT STATUS
010242 022700 047400 CMP #047400,FPS :CHECK FLOATING POINT STATUS
010246 001401 BEQ .+4 :BRANCH IF OK
010250 104000 HLT :FPS NOT EQUAL TO 047400

010252 022767 040000 170522 CMP #040000,ANS1 :DID 040000 GET STORED?
010260 001401 BEQ .+4 :BRANCH IF OK
010262 104004 HLT+4 :ANS1 NOT EQUAL TO 040000

010264 022767 000000 170512 CMP #000000,ANS2 :DID 000000 GET STORED?
010272 001401 BEQ .+4 :BRANCH IF OK
010274 104004 HLT+4 :ANS2 NOT EQUAL TO 000000

010276 005767 170504 TST ANS3 :DID ZERO GET STORED?
010302 001401 BEQ .+4 :BRANCH IF OK
010304 104004 HLT+4 :ANS3 NOT EQUAL TO ZERO

010306 005767 170476 TST ANS4 :DID ZERO GET STORED?
010312 001401 BEQ .+4 :BRANCH IF OK
010314 104004 HLT+4 :ANS4 NOT EQUAL TO ZERO

```

```

:*****
:TEST 67:      TEST STCFD (STORE-CONVERT FLOATING TO DOUBLE)
:      STORE  140000,000000 --> 140000,000000,000000,000000
:      FPS = 047410, AC = AC1,      DST = M6-R7
:*****

```

```

010316 104400      SCOPE
010320 000402      BR      TST67

010322 140000 000000 DAT67: 140000,000000

010326 170127 047417 TST67: LDFPS #047417 :LOAD FLOATING POINT STATUS

```

```

010332 172567 177764          LDF   DAT67, AC1      :LOAD 140000,000000 INTO AC1
010336 176167 170440      FPI67: STCFD   AC1,   ANS1      :STORE-CONVERT AC1 IN ANS1 THRU ANS4
010342 170200          STFPS  FPS          :STORE FLOATING POINT STATUS
010344 022700 047410      CMP    #047410,FPS   :CHECK FLOATING POINT STATUS
010350 001401          BEQ    .+4          :BRANCH IF OK
010352 104000          HLT                    :FPS NOT EQUAL TO 047410

010354 022767 140000 170420      CMP    #140000,ANS1  :DID 140000 GET STORED?
010362 001401          BEQ    .+4          :BRANCH IF OK
010364 104004          HLT+4        :ANS1 NOT EQUAL TO 140000

010366 022767 000000 170410      CMP    #000000,ANS2  :DID 000000 GET STORED?
010374 001401          BEQ    .+4          :BRANCH IF OK
010376 104004          HLT+4        :ANS2 NOT EQUAL TO 000000

010400 005767 170402          TST    ANS3          :DID ZERO GET STORED?
010404 001401          BEQ    .+4          :BRANCH IF OK
010406 104004          HLT+4        :ANS3 NOT EQUAL TO ZERO

010410 005767 170374          TST    ANS4          :DID ZERO GET STORED?
010414 001401          BEQ    .+4          :BRANCH IF OK
010416 104004          HLT+4        :ANS4 NOT EQUAL TO ZERO

```

```

*****
:TEST 70: TEST STCFD (STORE-CONVERT FLOATING TO DOUBLE)
:STORE 100000,000000 --> 000000,000000,000000,000000
:FPS = 047404, AC = AC3, DST = M6-R7
*****

```

```

010420 104400          SCOPE
010422 000402          BR      TST70

010424 100000 000000      DAT70: 100000,000000

010430 170127 000000      TST70: LDFPS   #0          :CLEAR FLOATING POINT STATUS
010434 172767 177764      LDF   DAT70, AC3      :LOAD 100000,000000 INTO AC3
010440 170127 047417      LDFPS #047417        :LOAD FLOATING POINT STATUS
010444 176367 170332      FPI70: STCFD   AC3,   ANS1      :STORE-CONVERT AC3 IN ANS1 THRU ANS4
010450 170200          STFPS  FPS          :STORE FLOATING POINT STATUS
010452 022700 047404      CMP    #047404,FPS   :CHECK FLOATING POINT STATUS
010456 001401          BEQ    .+4          :BRANCH IF OK
010460 104000          HLT                    :FPS NOT EQUAL TO 047404

010462 005767 170314          TST    ANS1          :CHECK ANS1
010466 001401          BEQ    .+4          :BRANCH IF OK
010470 104004          HLT+4        :ANS1 NOT EQUAL TO 100000

010472 005767 170306          TST    ANS2          :CHECK ANS2
010476 001401          BEQ    .+4          :BRANCH IF OK
010500 104004          HLT+4        :ANS2 NOT EQUAL TO 000000

010502 005767 170300          TST    ANS3          :DID ZERO GET STORED?
010506 001401          BEQ    .+4          :BRANCH IF OK
010510 104004          HLT+4        :ANS3 NOT EQUAL TO ZERO

```

```

010512 005767 170272 TST ANS4 ;DID ZERO GET STORED?
010516 001401 BEQ .+4 ;BRANCH IF OK
010520 104004 HLT+4 ;ANS4 NOT EQUAL TO ZERO

```

```

*****
:TEST 71: TEST STCFD (STORE-CONVERT FLOATING TO DOUBLE)
:STORE 177777,177777 --> 177777,177777,000000,000000
:FPS = 047410, AC = AC3, DST = M6-R7
*****

```

```

010522 104400 SCOPE
010524 000402 BR TST71

010526 177777 177777 DAT71: 177777,177777

010532 170127 047417 TST71: LDFPS #047417 ;LOAD FLOATING POINT STAU
010536 172767 177764 LDF DAT71, AC3 ;LOAD 177777,177777 INTO AC3
010542 176367 170234 FPI71: STCFD AC3, ANS1 ;STORE-CONVERT AC3 IN ANS1 THRU ANS4
010546 170200 STFPS FPS ;STORE FLOATING POINT STATUS
010550 022700 047410 CMP #047410,FPS ;CHECK FLOATING POINT STATUS
010554 001401 BEQ .+4 ;BRANCH IF OK
010556 104000 HLT ;FPS NOT EQUAL TO 047410

010560 022767 177777 170214 CMP #177777,ANS1 ;DID 177777 GET STORED?
010566 001401 BEQ .+4 ;BRANCH IF OK
010570 104004 HLT+4 ;ANS1 NOT EQUAL TO 177777

010572 022767 177777 170204 CMP #177777,ANS2 ;DID 177777 GET STORED?
010600 001401 BEQ .+4 ;BRANCH IF OK
010602 104004 HLT+4 ;ANS2 NOT EQUAL TO 177777

010604 005767 170176 TST ANS3 ;DID ZERO GET STORED?
010610 001401 BEQ .+4 ;BRANCH IF OK
010612 104004 HLT+4 ;ANS3 NOT EQUAL TO ZERO

010614 005767 170170 TST ANS4 ;DID ZERO GET STORED?
010620 001401 BEQ .+4 ;BRANCH IF OK
010622 104004 HLT+4 ;ANS4 NOT EQUAL TO ZERO

```

```

*****
:TEST 72: TEST STCFD (STORE-CONVERT FLOATING TO DOUBLE)
:STORE 040252,125252 --> 040252,125252,000000,000000
:FPS = 047400, AC = AC2, DST = M0-AC0
*****

```

```

010624 104400 SCOPE
010626 000402 BR TST72

010630 040252 125252 DAT72: 040252,125252

010634 170127 047417 TST72: LDFPS #047417 ;LOAD FLOATING POINT STAU
010640 172667 177764 LDF DAT72, AC2 ;LOAD 040252,125252 INTO AC2
010644 176200 STCFD AC2, AC0 ;STORE-CONVERT AC2 IN AC0
010646 170200 STFPS FPS ;STORE FLOATING POINT STATUS

```



```

011030 005767 167752      TST   ANS3      ;DID ZERO GET STORED?
011034 001401             BEQ   .+4       ;BRANCH IF OK
011036 104004             HLT+4          ;ANS3 NOT EQUAL TO ZERO

011040 005767 167744      TST   ANS4      ;DID ZERO GET STORED?
011044 001401             BEQ   .+4       ;BRANCH IF OK
011046 104004             HLT+4          ;ANS4 NOT EQUAL TO ZERO

```

```

*****
:TEST 74:      TEST STCFD (STORE-CONVERT FLOATING TO DOUBLE)
:              STORE  052525,052525 --> 052525,052525,000000,000000
:              FPS = 047400, AC = AC3, DST = M2-R2
*****

```

```

011050 104400      SCOPE
011052 000402      BR      TST74

011054 052525 052525  DAT74: 052525,052525

011060 170127 047417  TST74: LDFPS #047417 ;LOAD FLOATING POINT STAUS
011064 172767 177764   LDF   DAT74, AC3 ;LOAD 052525,052525 INTO AC3
011070 012702 001002   MOV   #ANS1, R2
011074 176322   STCFD AC3, (R2)+ ;STORE-CONVERT AC3 IN ANS1 THRU ANS4
011076 170200   STFPS FPS ;STORE FLOATING POINT STATUS
011100 022700 047400   CMP   #047400,FPS ;CHECK FLOATING POINT STATUS
011104 001401   BEQ   .+4 ;BRANCH IF OK
011106 104000   HLT ;FPS NOT EQUAL TO 047400

011110 022702 001012   CMP   #ANS5, R2 ;CHECK R2
011114 001401   BEQ   .+4 ;BRANCH IF OK
011116 104000   HLT ;R2 NOT EQUAL TO #ANS5

011120 022767 052525 167654  CMP   #052525,ANS1 ;DID 052525 GET STORED?
011126 001401   BEQ   .+4 ;BRANCH IF OK
011130 104004   HLT+4 ;ANS1 NOT EQUAL TO 052525

011132 022767 052525 167644  CMP   #052525,ANS2 ;DID 052525 GET STORED?
011140 001401   BEQ   .+4 ;BRANCH IF OK
011142 104004   HLT+4 ;ANS2 NOT EQUAL TO 052525

011144 005767 167636      TST   ANS3      ;DID ZERO GET STORED?
011150 001401             BEQ   .+4       ;BRANCH IF OK
011152 104004             HLT+4          ;ANS3 NOT EQUAL TO ZERO

011154 005767 167630      TST   ANS4      ;DID ZERO GET STORED?
011160 001401             BEQ   .+4       ;BRANCH IF OK
011162 104004             HLT+4          ;ANS4 NOT EQUAL TO ZERO

```

```

*****
:TEST 75:      TEST STCFD (STORE-CONVERT FLOATING TO DOUBLE)
:              STORE  125252,125252 --> 125252,125252,000000,000000
:              FPS = 047410, AC = AC0, DST = M3-R1
*****

```

```

011164 104400          SCOPE
011166 000403          BR      TST75

011170 125252 125252   DAT75: 125252,125252
011174 001002          ADR75: ANS1

011176 170127 047417   TST75: LDFPS #047417 ;LOAD FLOATING POINT STAUS
011202 012701 011174   MOV #ADR75, R1 ;PUT #ADR75 INTO R1
011206 172467 177756   LDF DAT75, ACO ;LOAD 125252,125252 INTO ACO
011212 176031          STCFD ACO, 2(R1)+ ;STORE-CONVERT ACO IN ANS1 THRU ANS4
011214 170200          STFPS FPS ;STORE FLOATING POINT STATUS
011216 022700 047410   CMP #047410,FPS ;CHECK FLOATING POINT STATUS
011222 001401          BEQ .+4 ;BRANCH IF OK
011224 104000          HLT ;FPS NOT EQUAL TO 047410

011226 022701 011176   CMP #TST75, R1 ;CHECK R1
011232 001401          BEQ .+4 ;BRANCH IF OK
011234 104000          HLT ;R1 NOT EQUAL TO #TST75

011236 022767 125252 167536 CMP #125252,ANS1 ;DID 125252 GET STORED?
011244 001401          BEQ .+4 ;BRANCH IF OK
011246 104004          HLT+4 ;ANS1 NOT EQUAL TO 125252

011250 022767 125252 167526 CMP #125252,ANS2 ;DID 125252 GET STORED?
011256 001401          BEQ .+4 ;BRANCH IF OK
011260 104004          HLT+4 ;ANS2 NOT EQUAL TO 125252

011262 005767 167520   TST ANS3 ;DID ZERO GET STORED?
011266 001401          BEQ .+4 ;BRANCH IF OK
011270 104004          HLT+4 ;ANS3 NOT EQUAL TO ZERO

011272 005767 167512   TST ANS4 ;DID ZERO GET STORED?
011276 001401          BEQ .+4 ;BRANCH IF OK
011300 104004          HLT+4 ;ANS4 NOT EQUAL TO ZERO

```

```

:*****
:TEST 76: TEST STCFD (STORE-CONVERT FLOATING TO DOUBLE)
:STORE 040252,125252 --> 040252,125252,000000,000000
:FPS = 047400, AC = ACO, DST = M4-R6
:*****

```

```

011302 104400          SCOPE
011304 000403          BR      TST76

011306 040252 125252   DAT76: 040252,125252
011312 000000          SAV76: 0

011314 170127 047417   TST76: LDFPS #047417 ;LOAD FLOATING POINT STAUS
011320 010667 177766   MOV %6, SAV76 ;SAVE R6
011324 172467 177756   LDF DAT76, ACO ;LOAD 040252,125252 INTO ACO
011330 176046          STCFD ACO, -(%6) ;STORE-CONVERT ACO IN ANS1 THRU ANS4
011332 170200          STFPS FPS ;STORE FLOATING POINT STATUS
011334 022700 047400   CMP #047400,FPS ;CHECK FLOATING POINT STATUS
011340 001401          BEQ .+4 ;BRANCH IF OK
011342 104000          HLT ;FPS NOT EQUAL TO 047400

```

```

011344 012667 167432      MOV      (%6)+, ANS1      ;POP THE ANSWER
011350 012667 167430      MOV      (%6)+, ANS2      ;
011354 012667 167426      MOV      (%6)+, ANS3      ;
011360 012667 167424      MOV      (%6)+, ANS4      ;
011364 020667 177722      CMP      %6, SAV76      ;CHECK R6
011370 001403      BEQ      TSA76          ;BRANCH IF OK
011372 010667 167414      MOV      %6, ANS5      ;SAVE WRONG %6 FOR TYPING
011376 104005      HLT+5      ;STACK POINTER FOULED UP, NOT 500

011400 022767 040252 167374 TSA76:  CMP      #040252,ANS1    ;DID 040252 GET STORED?
011406 001401      BEQ      .+4          ;BRANCH IF OK
011410 104004      HLT+4      ;ANS1 NOT EQUAL TO 040252

011412 022767 125252 167364      CMP      #125252,ANS2    ;DID 125252 GET STORED?
011420 001401      BEQ      .+4          ;BRANCH IF OK
011422 104004      HLT+4      ;ANS2 NOT EQUAL TO 125252

011424 005767 167356      TST      ANS3          ;DID ZERO GET STORED?
011430 001401      BEQ      .+4          ;BRANCH IF OK
011432 104004      HLT+4      ;ANS3 NOT EQUAL TO ZERO

011434 005767 167350      TST      ANS4          ;DID ZERO GET STORED?
011440 001401      BEQ      .+4          ;BRANCH IF OK
011442 104004      HLT+4      ;ANS4 NOT EQUAL TO ZERO

```

```

*****
:TEST 77:      TEST STCFD (STORE-CONVERT FLOATING TO DOUBLE)
:      STORE 140125,052525 --> 140125,052525,000000,000000
:      FPS = 047410, AC = ACO, DST = M5-R1
*****

```

```

011444 104400      SCOPE
011446 000403      BR      TST77

011450 140125 052525      DAT77: 140125,052525
011454 001002      ADR77: ANS1

011456 170127 047417      TST77: LDFPS      #047417      ;LOAD FLOATING POINT STAUS
011462 012701 011456      MOV      #TST77, R1      ;LOAD R1
011466 172467 177756      LDF      DAT77, ACO      ;LOAD 140125,052525 INTO ACO
011472 176051      STCFD   ACO, @-(R1)      ;STORE-CONVERT ACO IN ANS1 THRU ANS4
011474 170200      STFPS   FPS              ;STORE FLOATING POINT STATUS
011476 022700 047410      CMP      #047410,FPS      ;CHECK FLOATING POINT STATUS
011502 001401      BEQ      .+4          ;BRANCH IF OK
011504 104000      HLT      ;FPS NOT EQUAL TO 047410

011506 022701 011454      CMP      #ADR77, R1      ;CHECK R1
011512 001401      BEQ      .+4          ;BRANCH IF OK
011514 104000      HLT      ;R1 NOT EQUAL TO #ADR77

011516 022767 140125 167256      CMP      #140125,ANS1    ;DID 140125 GET STORED?
011524 001401      BEQ      .+4          ;BRANCH IF OK
011526 104004      HLT+4      ;ANS1 NOT EQUAL TO 140125

```

```

011530 022767 052525 167246      CMP      #052525,ANS2      ;DID 052525 GET STORED?
011536 001401                      BEQ      .+4              ;BRANCH IF OK
011540 104004                      HLT+4                    ;ANS2 NOT EQUAL TO 052525

011542 005767 167240      TST      ANS3              ;DID ZERO GET STORED?
011546 001401                      BEQ      .+4              ;BRANCH IF OK
011550 104004                      HLT+4                    ;ANS3 NOT EQUAL TO ZERO

011552 005767 167232      TST      ANS4              ;DID ZERO GET STORED?
011556 001401                      BEQ      .+4              ;BRANCH IF OK
011560 104004                      HLT+4                    ;ANS4 NOT EQUAL TO ZERO

```

```

:*****
:TEST 100:      TEST STCFD (STORE-CONVERT FLOATING TO DOUBLE)
:              STORE 040125,052525 --> 040125,052525,000000,000000
:              FPS = 047400, AC = AC1, DST = M6-R1
:*****

```

```

011562 104400      SCOPE
011564 000402      BR      TST100

011566 040125 052525      DAT100: 040125,052525

011572 170127 047417      TST100: LDFPS      #047417      ;LOAD FLOATING POINT STAUS
011576 012701 177645      MOV      #ANS1-1135,R1      ;LOAD R1
011602 172567 177760      LDF      DAT100,AC1        ;LOAD 040125,052525 INTO AC1
011606 176161 001135      STCFD    AC1,1135(R1)      ;STORE-CONVERT AC1 IN ANS1 THRU ANS4
011612 170200      STFPS    FPS              ;STORE FLOATING POINT STATUS
011614 022700 047400      CMP      #047400,FPS        ;CHECK FLOATING POINT STATUS
011620 001401                      BEQ      .+4              ;BRANCH IF OK
011622 104000                      HLT                    ;FPS NOT EQUAL TO 047400

011624 022701 177645      CMP      #ANS1-1135,R1      ;CHECK R1
011630 001401                      BEQ      .+4              ;BRANCH IF OK
011632 104000                      HLT                    ;R1 NOT EQUAL TO #ANS1-1135

011634 022767 040125 167140      CMP      #040125,ANS1      ;DID 040125 GET STORED?
011642 001401                      BEQ      .+4              ;BRANCH IF OK
011644 104004                      HLT+4                    ;ANS1 NOT EQUAL TO 040125

011646 022767 052525 167130      CMP      #052525,ANS2      ;DID 052525 GET STORED?
011654 001401                      BEQ      .+4              ;BRANCH IF OK
011656 104004                      HLT+4                    ;ANS2 NOT EQUAL TO 052525

011660 005767 167122      TST      ANS3              ;DID ZERO GET STORED?
011664 001401                      BEQ      .+4              ;BRANCH IF OK
011666 104004                      HLT+4                    ;ANS3 NOT EQUAL TO ZERO

011670 005767 167114      TST      ANS4              ;DID ZERO GET STORED?
011674 001401                      BEQ      .+4              ;BRANCH IF OK
011676 104004                      HLT+4                    ;ANS4 NOT EQUAL TO ZERO

```

```

:*****
:TEST 101:      TEST STCFD (STORE-CONVERT FLOATING TO DOUBLE)
:*****

```

: STORE 052525,052525 --> 052525,052525,000000,000000
: FPS = 047400, AC = AC0, DST = M7-R4
:*****

011700 104400
011702 000403

SCOPE
BR TST101

011704 052525 052525

DAT101: 052525,052525

011710 001002
011712 170127 047417
011716 012704 004007
011722 172467 177756
011726 176074 005701
011732 170200
011734 022700 047400
011740 001401
011742 104000

ADR101: ANS1
TST101: LDFPS #047417 :LOAD FLOATING POINT STAU
MOV #ADR101-5701,R4 :LOAD R4
LDF DAT101, AC0 :LOAD 052525,052525 INTO AC0
STCFD AC0, 25701(R4) :STORE-CONVERT AC0 IN ANS1 THRU ANS4
STFPS FPS :STORE FLOATING POINT STATUS
CMP #047400,FPS :CHECK FLOATING POINT STATUS
BEQ .+4 :BRANCH IF OK
HLT :FPS NOT EQUAL TO 047400

011744 022704 004007
011750 001401
011752 104000

CMP #ADR101-5701,R4 :CHECK R4
BEQ .+4 :BRANCH IF OK
HLT :R4 NOT EQUAL TO #ADR101-5701

011754 022767 052525 167020
011762 001401
011764 104004

CMP #052525,ANS1 :DID 052525 GET STORED?
BEQ .+4 :BRANCH IF OK
HLT+4 :ANS1 NOT EQUAL TO 052525

011766 022767 052525 167010
011774 001401
011776 104004

CMP #052525,ANS2 :DID 052525 GET STORED?
BEQ .+4 :BRANCH IF OK
HLT+4 :ANS2 NOT EQUAL TO 052525

012000 005767 167002
012004 001401
012006 104004

TST ANS3 :DID ZERO GET STORED?
BEQ .+4 :BRANCH IF OK
HLT+4 :ANS3 NOT EQUAL TO ZERO

012010 005767 166774
012014 001401
012016 104004

TST ANS4 :DID ZERO GET STORED?
BEQ .+4 :BRANCH IF OK
HLT+4 :ANS4 NOT EQUAL TO ZERO

:*****
:TEST 102: TEST STCFD (STORE-CONVERT FLOATING TO DOUBLE)
: STORE 125252,125252 --> 125252
: FPS = 047410, AC = AC3, DST = M2-R7
:*****

012020 104400
012022 000402

SCOPE
BR TST102

012024 125252 125252

DAT102: 125252,125252

012030 170127 047417
012034 172767 177764
012040 176327
012042 000000
012044 000402

TST102: LDFPS #047417 :LOAD FLOATING POINT STAU
LDF DAT102, AC3 :LOAD 125252,125252 INTO AC3
STCFD AC3, (PC)+ :STORE-CONVERT AC3 IN NEXT LOCATION
ADR102: 0 :LOCATION OF ANSWER
BR .+6

```

012046 000000 HALT
012050 000000 HALT
012052 016767 177764 166722 MOV ADR102,ANS1
012060 016767 177760 166716 MOV ADR102+2,ANS2
012066 016767 177754 166714 MOV ADR102+4,ANS4
012074 016767 177750 166712 MOV ADR102+6,ANS6
012102 170200 STFPS FPS :STORE FLOATING POINT STATUS
012104 022700 047410 CMP #047410,FPS :CHECK FLOATING POINT STATUS
012110 001401 BEQ .+4 :BRANCH IF OK
012112 104000 HLT :FPS NOT EQUAL TO 047410

012114 022767 125252 166660 CMP #125252,ANS1 :DID 125252 GET STORED?
012122 001401 BEQ .+4 :BRANCH IF OK
012124 104004 HLT+4 :ANS1 NOT EQUAL TO 125252

012126 022767 000402 166550 CMP #402, ANS2 :DID "BR .+6" GET STORED?
012134 001401 BEQ .+4 :BRANCH IF OK
012136 104004 HLT+4 :ANS2 NOT EQUAL TO "BR .+6"

012140 005767 166642 TST ANS3 :DID ZERO GET STORED?
012144 001401 BEQ .+4 :BRANCH IF OK
012146 104004 HLT+4 :ANS3 NOT EQUAL TO ZERO

012150 005767 166634 TST ANS4 :DID ZERO GET STORED?
012154 001401 BEQ .+4 :BRANCH IF OK
012156 104004 HLT+4 :ANS4 NOT EQUAL TO ZERO

```

```

*****
:TEST 103: TEST STCFD (STORE-CONVERT FLOATING TO DOUBLE)
: STORE 040125,052525 --> 040125,052525,000000,000000
: FPS = 047400, AC = AC2, DST = M3-R7
*****

```

```

012160 104400 SCOPE
012162 000402 BR TST103

012164 040125 052525 DAT103: 040125,052525

012170 170127 047417 TST103: LDFPS #047417 :LOAD FLOATING POINT STATUS
012174 172667 177764 LDF DAT103, AC2 :LOAD 040125,052525 INTO AC2
012200 176237 001002 STCFD AC2, 2#ANS1 :STORE-CONVERT AC2 IN ANS1 THRU ANS4
012204 170200 STFPS FPS :STORE FLOATING POINT STATUS
012206 022700 047400 CMP #047400,FPS :CHECK FLOATING POINT STATUS
012212 001401 BEQ .+4 :BRANCH IF OK
012214 104000 HLT :FPS NOT EQUAL TO 047400

012216 022767 040125 166556 CMP #040125,ANS1 :DID 040125 GET STORED?
012224 001401 BEQ .+4 :BRANCH IF OK
012226 104004 HLT+4 :ANS1 NOT EQUAL TO 040125

012230 022767 052525 166546 CMP #052525,ANS2 :DID 052525 GET STORED?
012236 001401 BEQ .+4 :BRANCH IF OK
012240 104004 HLT+4 :ANS2 NOT EQUAL TO 052525

012242 005767 166540 TST ANS3 :DID ZERO GET STORED?

```

MAINDEC-11-DCFPI-B
DCFPI.P11

TEST OF LDCDF, LDCFD, STCFD, STCDF
TEST SECTION

012246	001401		BEG	.+4	:BRANCH IF OK
012250	104004		HLT+4		:ANS3 NOT EQUAL TO ZERO
012252	005767	166532	TST	ANS4	:DID ZERO GET STORED?
012256	001401		BEG	.+4	:BRANCH IF OK
012260	104004		HLT+4		:ANS4 NOT EQUAL TO ZERO

```

*****
:TEST 104: TEST STCFD (STORE-CONVERT FLOATING TO DOUBLE)
:STORE 140252,125252 --> 140252,125252,000000,000000
:FPS = 047410, AC = AC3, DST = M7-R7
*****

```

012262	104400		SCOPE		
--------	--------	--	-------	--	--

012264	000403		BR	TST104	
--------	--------	--	----	--------	--

012266	140252	125252	DAT104:	140252,125252	
012272	001002		ADR104:	ANS1	

012274	170127	047417	TST104:	LDFPS	#047417	:LOAD FLOATING POINT STATUS
012300	172767	177762		LDF	DAT104, AC3	:LOAD 140252,125252 INTO AC3
012304	176377	177762		STCFD	AC3, BADR104	:STORE-CONVERT AC3 IN ANS1 THRU ANS4
012310	170200			STFPS	FPS	:STORE FLOATING POINT STATUS
012312	022700	047410		CMP	#047410,FPS	:CHECK FLOATING POINT STATUS
012316	001401			BEG	.+4	:BRANCH IF OK
012320	104000			HLT		:FPS NOT EQUAL TO 047410

012322	022767	140252	166452	CMP	#140252,ANS1	:DID 140252 GET STORED?
012330	001401			BEG	.+4	:BRANCH IF OK
012332	104004			HLT+4		:ANS1 NOT EQUAL TO 140252

012334	022767	125252	166442	CMP	#125252,ANS2	:DID 125252 GET STORED?
012342	001401			BEG	.+4	:BRANCH IF OK
012344	104004			HLT+4		:ANS2 NOT EQUAL TO 125252

012346	005767	166434		TST	ANS3	:DID ZERO GET STORED?
012352	001401			BEG	.+4	:BRANCH IF OK
012354	104004			HLT+4		:ANS3 NOT EQUAL TO ZERO

012356	005767	166426		TST	ANS4	:DID ZERO GET STORED?
012362	001401			BEG	.+4	:BRANCH IF OK
012364	104004			HLT+4		:ANS4 NOT EQUAL TO ZERO

```

*****
:TEST 105: TEST STCDF (STORE-CONVERT DOUBLE TO FLOATING)
:STORE 000000,000000,000000,000000 --> 000000,000000
:FPS = 047604, AC = AC0, DST = M6-R7
*****

```


MAINDEC-11-DOFPI-B
DOFPI.P11

TEST OF LDCDF, LDCFD, STCFD, STCDF
TEST SECTION

```

012366 104400          SCOPE
012370 000404          BR      TST105

012372 000000 000000 000000 DAT105: 000000,000000,000000,000000
012400 000000

012402 170127 047617   TST105: LDFPS  #047617   :LOAD FLOATING POINT STAUS
012406 005067 166374   CLR      ANS3   :CLEAR ANS3
012412 005067 166372   CLR      ANS4   :CLEAR ANS4
012416 172467 177750   LDD     DAT105, ACO :LOAD 000000,000000,000000,000000 INTO ACO
012422 176067 166354   FPI105: STCDF  ACO,  ANS1 :STORE-CONVERT ACO IN ANS1, ANS2
012426 170200 170200   STFPS   FPS     :STORE FLOATING POINT STATUS
012430 022700 047604   CMP     #047604,FPS :CHECK FLOATING POINT STATUS
012434 001401 001401   BEQ     .+4      :BRANCH IF OK
012436 104000 104000   HLT

012440 022767 000000 166334   CMP     #000000,ANS1 :DID 000000 GET STORED?
012446 001401 001401   BEQ     .+4      :BRANCH IF OK
012450 104004 104004   HLT+4   :ANS1 NOT EQUAL TO 000000

012452 022767 000000 166324   CMP     #000000,ANS2 :DID 000000 GET STORED?
012460 001401 001401   BEQ     .+4      :BRANCH IF OK
012462 104004 104004   HLT+4   :ANS2 NOT EQUAL TO 000000

012464 005767 166316   TST     ANS3     :DID ZERO GET STORED?
012470 001401 001401   BEQ     .+4      :BRANCH IF OK
012472 104004 104004   HLT+4   :ANS3 NOT EQUAL TO ZERO

012474 005767 166310   TST     ANS4     :DID ZERO GET STORED?
012500 001401 001401   BEQ     .+4      :BRANCH IF OK
012502 104004 104004   HLT+4   :ANS4 NOT EQUAL TO ZERO

```

```

:*****
:TEST 106: TEST STCDF (STORE-CONVERT DOUBLE TO FLOATING)
: STORE 052525,052525,052525,052525 --> 052525,052525
: FPS = 047600, AC = ACO, DST = M6-R7
:*****

```

```

012504 104400          SCOPE
012506 000404          BR      TST106

012510 052525 052525 052525 DAT106: 052525,052525,052525,052525
012516 052525

012520 170127 047617   TST106: LDFPS  #047617   :LOAD FLOATING POINT STAUS
012524 005067 166256   CLR      ANS3   :CLEAR ANS3
012530 005067 166254   CLR      ANS4   :CLEAR ANS4
012534 172467 177750   LDD     DAT106, ACO :LOAD 052525,052525,052525,052525 INTO ACO
012540 176067 166236   FPI106: STCDF  ACO,  ANS1 :STORE-CONVERT ACO IN ANS1, ANS2
012544 170200 170200   STFPS   FPS     :STORE FLOATING POINT STATUS
012546 022700 047600   CMP     #047600,FPS :CHECK FLOATING POINT STATUS
012552 001401 001401   BEQ     .+4      :BRANCH IF OK
012554 104000 104000   HLT

012556 022767 052525 166216   CMP     #052525,ANS1 :DID 052525 GET STORED?

```

```

012564 001401      BEQ      .+4      :BRANCH IF OK
012566 104004      HLT+4      :ANS1 NOT EQUAL TO 052525

012570 022767 052525 166206  CMP      #052525,ANS2 :DID 052525 GET STORED?
012576 001401      BEQ      .+4      :BRANCH IF OK
012600 104004      HLT+4      :ANS2 NOT EQUAL TO 052525

012602 005767 166200      TST      ANS3      :DID ZERO GET STORED?
012606 001401      BEQ      .+4      :BRANCH IF OK
012610 104004      HLT+4      :ANS3 NOT EQUAL TO ZERO

012612 005767 166172      TST      ANS4      :DID ZERO GET STORED?
012616 001401      BEQ      .+4      :BRANCH IF OK
012620 104004      HLT+4      :ANS4 NOT EQUAL TO ZERO

```

```

:*****
:TEST 107: TEST STCDF (STORE-CONVERT DOUBLE TO FLOATING)
: STORE 125252,125252,125252,125252 --> 125252,125253
: FPS = 047610, AC = AC2, DST = M-R7
:*****

```

```

012622 104400      SCOPE
012624 000404      BR      TST107

012626 125252 125252 125252 DAT107: 125252,125252,125252,125252
012634 125252

012636 170127 047617 TST107: LDFPS #047617 :LOAD FLOATING POINT STATUS
012642 005067 166140 CLR ANS3 :CLEAR ANS3
012646 005067 166136 CLR ANS4 :CLEAR ANS4
012652 172667 177750 LDD DAT107, AC2 :LOAD 125252,125252,125252,125252 INTO AC2
012656 176267 166120 FPI107: STCDF AC2, ANS1 :STORE-CONVERT AC2 IN ANS1, ANS2
012662 170200 STFPS FPS :STORE FLOATING POINT STATUS
012664 022700 047610 CMP #047610,FPS :CHECK FLOATING POINT STATUS
012670 001401 BEQ .+4 :BRANCH IF OK
012672 104000 HLT :FPS NOT EQUAL TO 047610

012674 022767 125252 166100 CMP #125252,ANS1 :DID 125252 GET STORED?
012702 001401 BEQ .+4 :BRANCH IF OK
012704 104004 HLT+4 :ANS1 NOT EQUAL TO 125252

012706 022767 125253 166070 CMP #125253,ANS2 :DID 125253 GET STORED?
012714 001401 BEQ .+4 :BRANCH IF OK
012716 104004 HLT+4 :ANS2 NOT EQUAL TO 125253

012720 005767 166062 TST ANS3 :DID ZERO GET STORED?
012724 001401 BEQ .+4 :BRANCH IF OK
012726 104004 HLT+4 :ANS3 NOT EQUAL TO ZERO

012730 005767 166054 TST ANS4 :DID ZERO GET STORED?
012734 001401 BEQ .+4 :BRANCH IF OK
012736 104004 HLT+4 :ANS4 NOT EQUAL TO ZERO

```

```

:*****

```

:TEST 110: TEST STCDF (STORE-CONVERT DOUBLE TO FLOATING)
: STORE 000200,000000,000000,000000 --> 000200,000000
: FPS = 047600, AC = AC2, DST = M6-R7
:*****

012740	104400			SCOPE			
012742	000404			BR	TST110		
012744	000200	000000	000000	DAT110:	000200,000000,000000,000000		
012752	000000						
012754	170127	047617		TST110:	LDFPS #047617	:LOAD FLOATING POINT STAUS	
012760	005067	166022			CLR ANS3	:CLEAR ANS3	/
012764	005067	166020			CLR ANS4	:CLEAR ANS4	
012770	172667	177750			LDD DAT110, AC2	:LOAD 000200,000000,000000,000000 INTO AC2	
012774	176267	166002		FPI110:	STCDF AC2, ANS1	:STORE-CONVERT AC2 IN ANS1, ANS2	
013000	170200				STFPS FPS	:STORE FLOATING POINT STATUS	
013002	022700	047600			CMP #047600,FPS	:CHECK FLOATING POINT STATUS	
013006	001401				BEG .+4	:BRANCH IF OK	
013010	104000				HLT	:FPS NOT EQUAL TO 047600	
013012	022767	000200	165762		CMP #000200,ANS1	:DID 000200 GET STORED?	
013020	001401				BEG .+4	:BRANCH IF OK	
013022	104004				HLT+4	:ANS1 NOT EQUAL TO 000200	
013024	022767	000000	165752		CMP #000000,ANS2	:DID 000000 GET STORED?	
013032	001401				BEG .+4	:BRANCH IF OK	
013034	104004				HLT+4	:ANS2 NOT EQUAL TO 000000	
013036	005767	165744			TST ANS3	:DID ZERO GET STORED?	
013042	001401				BEG .+4	:BRANCH IF OK	
013044	104004				HLT+4	:ANS3 NOT EQUAL TO ZERO	
013046	005767	165736			TST ANS4	:DID ZERO GET STORED?	
013052	001401				BEG .+4	:BRANCH IF OK	
013054	104004				HLT+4	:ANS4 NOT EQUAL TO ZERO	

:*****
:TEST 111: TEST STCDF (STORE-CONVERT DOUBLE TO FLOATING)
: STORE 100200,000000,000000,000000 --> 100200,000000
: FPS = 047610, AC = AC3, DST = M6-R7
:*****

013056	104400			SCOPE			
013060	000404			BR	TST111		
013062	100200	000000	000000	DAT111:	100200,000000,000000,000000		
013070	000000						
013072	170127	047617		TST111:	LDFPS #047617	:LOAD FLOATING POINT STAUS	
013076	005067	165704			CLR ANS3	:CLEAR ANS3	
013102	005067	165702			CLR ANS4	:CLEAR ANS4	
013106	172767	177750			LDD DAT111, AC3	:LOAD 100200,000000,000000,000000 INTO AC3	
013112	176367	165664		FPI111:	STCDF AC3, ANS1	:STORE-CONVERT AC3 IN ANS1, ANS2	
013116	170200				STFPS FPS	:STORE FLOATING POINT STATUS	

```

013120 022700 047610      CMP      #047610,FPS      :CHECK FLOATING POINT STATUS
013124 001401      BEQ      .+4          :BRANCH IF OK
013126 104000      HLT                      :FPS NOT EQUAL TO 047610

013130 022767 100200 165644  CMP      #100200,ANS1   :DID 100200 GET STORED?
013136 001401      BEQ      .+4          :BRANCH IF OK
013140 104004      HLT+4        :ANS1 NOT EQUAL TO 100200

013142 022767 000000 165634  CMP      #000000,ANS2   :DID 000000 GET STORED?
013150 001401      BEQ      .+4          :BRANCH IF OK
013152 104004      HLT+4        :ANS2 NOT EQUAL TO 000000

013154 005767 165626      TST      ANS3          :DID ZERO GET STORED?
013160 001401      BEQ      .+4          :BRANCH IF OK
013162 104004      HLT+4        :ANS3 NOT EQUAL TO ZERO

013164 005767 165620      TST      ANS4          :DID ZERO GET STORED?
013170 001401      BEQ      .+4          :BRANCH IF OK
013172 104004      HLT+4        :ANS4 NOT EQUAL TO ZERO

```

```

:*****
:TEST 112:      TEST STCDF (STORE-CONVERT DOUBLE TO FLOATING)
:      STORE 000177,177777,177777,177777 --> 000000,000000
:      FPS = 047604, AC = AC1, DST = M6-R7
:*****

```

```

013174 104400      SCOPE
013176 000404      BR      TST112

013200 000177 177777 177777 DAT112: 000177,177777,177777,177777
013206 177777

013210 170127 000200      TST112: LDFPS      #000200      :SETD ONLY
013214 005067 165566      CLR      ANS3          :CLEAR ANS3
013220 005067 165564      CLR      ANS4          :CLEAR ANS4
013224 172567 177750      LDD      DAT112, AC1   :LOAD 000177,177777,177777,177777 INTO AC1
013230 170127 047617      LDFPS      #047617     :LOAD FLOATING POINT STATUS
013234 176167 165542      FPI112: STCDF      AC1, ANS1 :STORE-CONVERT AC1 IN ANS1, ANS2
013240 170200      STFPS      FPS          :STORE FLOATING POINT STATUS
013242 022700 047604      CMP      #047604,FPS   :CHECK FLOATING POINT STATUS
013246 001401      BEQ      .+4          :BRANCH IF OK
013250 104000      HLT                      :FPS NOT EQUAL TO 047604

013252 005767 165524      TST      ANS1          :CHECK ANS1
013256 001401      BEQ      .+4          :BRANCH IF OK
013260 104004      HLT+4        :ANS1 NOT EQUAL TO E

013262 005767 165516      TST      ANS2          :CHECK ANS2
013266 001401      BEQ      .+4          :BRANCH IF OK
013270 104004      HLT+4        :ANS2 NOT EQUAL TO F

013272 005767 165510      TST      ANS3          :DID ZERO GET STORED?
013276 001401      BEQ      .+4          :BRANCH IF OK
013300 104004      HLT+4        :ANS3 NOT EQUAL TO ZERO

```

```

013302 005767 165502      TST      ANS4      ;DID ZERO GET STORED?
013306 001401      BEQ      .+4      ;BRANCH IF OK
013310 104004      HLT+4     ;ANS4 NOT EQUAL TO ZERO

```

```

:*****
:TEST 113:      TEST STCDF (STORE-CONVERT DOUBLE TO FLOATING)
:      STORE  040000,000000,000000,000000 --> 040000,000000
:      FPS = 047600, AC = AC3, DST = M6-R7
:*****

```

```

013312 104400      SCOPE
013314 000404      BR      TST113

```

```

013316 040000 000000 000000 DAT113: 040000,000000,000000,000000
013324 000000

```

```

013326 170127 047617      TST113: LDFPS  #047617      ;LOAD FLOATING POINT STATUS
013332 005067 165450      CLR      ANS3      ;CLEAR ANS3
013336 005067 165446      CLR      ANS4      ;CLEAR ANS4
013342 172767 177750      LDD      DAT113, AC3 ;LOAD 040000,000000,000000,000000 INTO AC3
013346 176367 165430      FPI113: STCDF  AC3, ANS1 ;STORE-CONVERT AC3 IN ANS1, ANS2
013352 170200      STFPS   FPS      ;STORE FLOATING POINT STATUS
013354 022700 047600      CMP      #047600,FPS ;CHECK FLOATING POINT STATUS
013360 001401      BEQ      .+4      ;BRANCH IF OK
013362 104000      HLT     ;FPS NOT EQUAL TO 047600

```

```

013364 022767 040000 165410      CMP      #040000,ANS1 ;DID 040000 GET STORED?
013372 001401      BEQ      .+4      ;BRANCH IF OK
013374 104004      HLT+4     ;ANS1 NOT EQUAL TO 040000

```

```

013376 022767 000000 165400      CMP      #000000,ANS2 ;DID 000000 GET STORED?
013404 001401      BEQ      .+4      ;BRANCH IF OK
013406 104004      HLT+4     ;ANS2 NOT EQUAL TO 000000

```

```

013410 005767 165372      TST      ANS3      ;DID ZERO GET STORED?
013414 001401      BEQ      .+4      ;BRANCH IF OK
013416 104004      HLT+4     ;ANS3 NOT EQUAL TO ZERO

```

```

013420 005767 165364      TST      ANS4      ;DID ZERO GET STORED?
013424 001401      BEQ      .+4      ;BRANCH IF OK
013426 104004      HLT+4     ;ANS4 NOT EQUAL TO ZERO

```

```

:*****
:TEST 114:      TEST STCDF (STORE-CONVERT DOUBLE TO FLOATING)
:      STORE  140000,000000,100000,000000 --> 140000,000001
:      FPS = 047610, AC = AC1, DST = M6-R7
:*****

```

```

013430 104400      SCOPE
013432 000404      BR      TST114

```

```

013434 140000 000000 100000 DAT114: 140000,000000,100000,000000
013442 000000

```

```

013444 170127 047617 TST114: LDFPS #047617 ;LOAD FLOATING POINT STAUS
013450 005067 165332 CLR ANS3 ;CLEAR ANS3
013454 005067 165330 CLR ANS4 ;CLEAR ANS4
013460 172567 177750 LDD DAT114, AC1 ;LOAD 140000,000000,100000,000000 INTO AC1
013464 176167 165312 FPI114: STCDF AC1, ANS1 ;STORE-CONVERT AC1 IN ANS1, ANS2
013470 170200 STFPS FPS ;STORE FLOATING POINT STATUS
013472 022700 047610 CMP #047610,FPS ;CHECK FLOATING POINT STATUS
013476 001401 BEQ .+4 ;BRANCH IF OK
013500 104000 HLT ;FPS NOT EQUAL TO 047610

013502 022767 140000 165272 CMP #140000,ANS1 ;DID 140000 GET STORED?
013510 001401 BEQ .+4 ;BRANCH IF OK
013512 104004 HLT+4 ;ANS1 NOT EQUAL TO 140000

013514 022767 000001 165262 CMP #000001,ANS2 ;DID 000001 GET STORED?
013522 001401 BEQ .+4 ;BRANCH IF OK
013524 104004 HLT+4 ;ANS2 NOT EQUAL TO 000001

013526 005767 165254 TST ANS3 ;DID ZERO GET STORED?
013532 001401 BEQ .+4 ;BRANCH IF OK
013534 104004 HLT+4 ;ANS3 NOT EQUAL TO ZERO

013536 005767 165246 TST ANS4 ;DID ZERO GET STORED?
013542 001401 BEQ .+4 ;BRANCH IF OK
013544 104004 HLT+4 ;ANS4 NOT EQUAL TO ZERO

```

```

*****
:TEST 115: TEST STCDF (STORE-CONVERT DOUBLE TO FLOATING)
: STORE 177777,177777,077777,177777 --> 177777,177777
: FPS = 047610, AC = AC1, DST = M6-R7
*****

```

```

013546 104400 SCOPE
013550 000404 BR TST115

013552 177777 177777 077777 DAT115: 177777,177777,077777,177777
013560 177777

013562 170127 047617 TST115: LDFPS #047617 ;LOAD FLOATING POINT STAUS
013566 005067 165214 CLR ANS3 ;CLEAR ANS3
013572 005067 165212 CLR ANS4 ;CLEAR ANS4
013576 172567 177750 LDD DAT115, AC1 ;LOAD 177777,177777,077777,177777 INTO AC1
013602 176167 165174 FPI115: STCDF AC1, ANS1 ;STORE-CONVERT AC1 IN ANS1, ANS2
013606 170200 STFPS FPS ;STORE FLOATING POINT STATUS
013610 022700 047610 CMP #047610,FPS ;CHECK FLOATING POINT STATUS
013614 001401 BEQ .+4 ;BRANCH IF OK
013616 104000 HLT ;FPS NOT EQUAL TO 047610

013620 022767 177777 165154 CMP #177777,ANS1 ;DID 177777 GET STORED?
013626 001401 BEQ .+4 ;BRANCH IF OK
013630 104004 HLT+4 ;ANS1 NOT EQUAL TO 177777

013632 022767 177777 165144 CMP #177777,ANS2 ;DID 177777 GET STORED?
013640 001401 BEQ .+4 ;BRANCH IF OK
013642 104004 HLT+4 ;ANS2 NOT EQUAL TO 177777

```

```

013644 005767 165136      TST      ANS3      ;DID ZERO GET STORED?
013650 001401      BEQ      .+4      ;BRANCH IF OK
013652 104004      HLT+4      ;ANS3 NOT EQUAL TO ZERO

013654 005767 165130      TST      ANS4      ;DID ZERO GET STORED?
013660 001401      BEQ      .+4      ;BRANCH IF OK
013662 104004      HLT+4      ;ANS4 NOT EQUAL TO ZERO

```

```

:*****
:TEST 116:      TEST STCDF (STORE-CONVERT DOUBLE TO FLOATING)
:      STORE 177777,177777,177777,177777 --> 100000,000000
:      FPS = 147616, AC = AC1, DST = M6-R7
:      FEC = 10, FEA = FPI116
:*****

```

```

013664 104400      SCOPE
013666 000404      BR      TST116

013670 177777 177777 177777 DAT116: 177777,177777,177777,177777
013676 177777

013700 170127 047617      TST116: LDFPS #047617 ;LOAD FLOATING POINT STATUS
013704 005067 165076      CLR      ANS3      ;CLEAR ANS3
013710 005067 165074      CLR      ANS4      ;CLEAR ANS4
013714 172567 177750      LDD      DAT116, AC1 ;LOAD 177777,177777,177777,177777 INTO AC1
013720 176167 165056      FPI116: STCDF AC1, ANS1 ;STORE-CONVERT AC1 IN ANS1, ANS2
013724 170200      STFPS    FPS      ;STORE FLOATING POINT STATUS
013726 170367 165070      STST     FEC      ;STORE EXCEPTION CODES
013732 022700 147616      CMP      #147616,FPS ;CHECK FLOATING POINT STATUS
013736 001401      BEQ      .+4      ;BRANCH IF OK
013740 104000      HLT      ;FPS NOT EQUAL TO 147616

013742 022767 000010 165052      CMP      #10, FEC   ;CHECK FLOATING EXCEPTION CODE
013750 001401      BEQ      .+4      ;BRANCH IF OK
013752 104000      HLT      ;FEC NOT EQUAL TO 10

013754 022767 013720 165042      CMP      #FPI116, FEA ;CHECK FLOATING EXCEPTION ADDRESS
013762 001401      BEQ      .+4      ;BRANCH IF OK
013764 104000      HLT      ;FEA NOT EQUAL TO FPI116

013766 022767 100000 165006      CMP      #100000,ANS1 ;DID 100000 GET STORED?
013774 001401      BEQ      .+4      ;BRANCH IF OK
013776 104004      HLT+4     ;ANS1 NOT EQUAL TO 100000

014000 022767 000000 164776      CMP      #000000,ANS2 ;DID 000000 GET STORED?
014006 001401      BEQ      .+4      ;BRANCH IF OK
014010 104004      HLT+4     ;ANS2 NOT EQUAL TO 000000

014012 005767 164770      TST      ANS3      ;DID ZERO GET STORED?
014016 001401      BEQ      .+4      ;BRANCH IF OK
014020 104004      HLT+4     ;ANS3 NOT EQUAL TO ZERO

014022 005767 164762      TST      ANS4      ;DID ZERO GET STORED?
014026 001401      BEQ      .+4      ;BRANCH IF OK

```

014030 104004

HLT+4

;ANS4 NOT EQUAL TO ZERO

```

*****
:TEST 117: TEST STCDF (STORE-CONVERT DOUBLE TO FLOATING)
:          STORE 077777,177777,177777,177777 --> 000000,000000
:          FPS = 147606, AC = AC2, DST = M6-R7
:          FEC = 10, FEA = FPI117
*****

```

014032 104400
014034 000404

SCOPE
BR TST117

014036 077777 177777 177777 DAT117: 077777,177777,177777,177777
014044 177777

014046 170127 047617
014052 005067 164730
014056 005067 164726
014062 172667 177750
014066 176267 164710
014072 170200
014074 170367 164722
014100 022700 147606
014104 001401
014106 104000

```

TST117: LDFPS #047617 ;LOAD FLOATING POINT STAUS
          CLR ANS3 ;CLEAR ANS3
          CLR ANS4 ;CLEAR ANS4
          LDD DAT117, AC2 ;LOAD 077777,177777,177777,177777 INTO AC2
FPI117: STCDF AC2, ANS1 ;STORE-CONVERT AC2 IN ANS1, ANS2
          STFPS FPS ;STORE FLOATING POINT STATUS
          STST FEC ;STORE EXCEPTION CODES
          CMP #147606, FPS ;CHECK FLOATING POINT STATUS
          BEQ .+4 ;BRANCH IF OK
          HLT ;FPS NOT EQUAL TO 147606

```

014110 022767 000010 164704
014116 001401
014120 104000

```

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

```

014122 022767 014066 164674
014130 001401
014132 104000

```

CMP #FPI117, FEA ;CHECK FLOATING EXCEPTION ADDRESS
BEQ .+4 ;BRANCH IF OK
HLT ;FEA NOT EQUAL TO FPI117

```

014134 022767 000000 164640
014142 001401
014144 104004

```

CMP #000000, ANS1 ;DID 000000 GET STORED?
BEQ .+4 ;BRANCH IF OK
HLT+4 ;ANS1 NOT EQUAL TO 000000

```

014146 022767 000000 164630
014154 001401
014156 104004

```

CMP #000000, ANS2 ;DID 000000 GET STORED?
BEQ .+4 ;BRANCH IF OK
HLT+4 ;ANS2 NOT EQUAL TO 000000

```

014160 005767 164622
014164 001401
014166 104004

```

TST ANS3 ;DID ZERO GET STORED?
BEQ .+4 ;BRANCH IF OK
HLT+4 ;ANS3 NOT EQUAL TO ZERO

```

014170 005767 164614
014174 001401
014176 104004

```

TST ANS4 ;DID ZERO GET STORED?
BEQ .+4 ;BRANCH IF OK
HLT+4 ;ANS4 NOT EQUAL TO ZERO

```

```

*****
:TEST 120: TEST STCDF (STORE-CONVERT DOUBLE TO FLOATING)
:          STORE 040252,125252,125252,125252 --> 040252,125253
:          FPS = 047600, AC = AC0, DST = M0-AC3
:

```

```

014200 104400          SCOPE
014202 000404          BR      TST120

014204 040252 125252 125252 DAT120: 040252,125252,125252,125252
014212 125252

014214 170127 047617    TST120: LDFPS  #047617      ;LOAD FLOATING POINT STAUS
014220 005067 164562    CLR      ANS3      ;CLEAR ANS3
014224 005067 164560    CLR      ANS4      ;CLEAR ANS4
014230 172467 177750    LDD     DAT120, ACO ;LOAD 040252,125252,125252,125252 INTO ACO
014234 176003  STCDF  ACO,   AC3  ;STORE-CONVERT ACO INTO AC3
014236 170200  STFPS  FPS      ;STORE FLOATING POINT STATUS
014240 022700 047600    CMP     #047600,FPS ;CHECK FLOATING POINT STATUS
014244 001401  BEQ     .+4       ;BRANCH IF OK
014246 104000  HLT                    ;FPS NOT EQUAL TO 047600

014250 174367 164526 164520 STD     AC3,   ANS1  ;STORE AC3 IN ANS1, ANS2
014254 022767 040252 164520 CMP     #040252,ANS1 ;DID 040252 GET STORED?
014262 001401  BEQ     .+4       ;BRANCH IF OK
014264 104004  HLT+4      ;ANS1 NOT EQUAL TO 040252

014266 022767 125253 164510 CMP     #125253,ANS2 ;DID 125253 GET STORED?
014274 001401  BEQ     .+4       ;BRANCH IF OK
014276 104004  HLT+4      ;ANS2 NOT EQUAL TO 125253

014300 005767 164502  TST     ANS3      ;DID ZERO GET STORED?
014304 001401  BEQ     .+4       ;BRANCH IF OK
014306 104004  HLT+4      ;ANS3 NOT EQUAL TO ZERO

014310 005767 164474  TST     ANS4      ;DID ZERO GET STORED?
014314 001401  BEQ     .+4       ;BRANCH IF OK
014316 104004  HLT+4      ;ANS4 NOT EQUAL TO ZERO

```

```

*****
;TEST 121: TEST STCDF (STORE-CONVERT DOUBLE TO FLOATING)
;STORE 052525,052525,052525,052525 --> 052525,052525
;FPS = 047600, AC = ACO, DST = M1-R4
*****

```

```

014320 104400          SCOPE
014322 000404          BR      TST121

014324 052525 052525 052525 DAT121: 052525,052525,052525,052525
014332 052525
; \#052525,052525,052525,052525

014334 170127 047617    TST121: LDFPS  #047617      ;LOAD FLOATING POINT STAUS
014340 005067 164442    CLR      ANS3      ;CLEAR ANS3
014344 005067 164440    CLR      ANS4      ;CLEAR ANS4
014350 172467 177750    LDD     DAT121, ACO ;LOAD 052525,052525,052525,052525 INTO ACO
014354 012704 001002    MOV     #ANS1, R4
014360 176014  STCDF  ACO,   (R4) ;STORE-CONVERT ACO IN ANS1, ANS2
014362 170200  STFPS  FPS      ;STORE FLOATING POINT STATUS

```

```

014364 022700 047600      CMP      #047600,FPS      ;CHECK FLOATING POINT STATUS
014370 001401      BEQ      .+4            ;BRANCH IF OK
014372 104000      HLT                               ;FPS NOT EQUAL TO 047600

014374 022704 001002      CMP      #ANS1, R4      ;CHECK R4
014400 000401      BR      .+4            ;BRANCH IF OK
014402 104000      HLT                               ;R4 NOT EQUAL TO #ANS1

014404 022767 052525 164370  CMP      #052525,ANS1   ;DID 052525 GET STORED?
014412 001401      BEQ      .+4            ;BRANCH IF OK
014414 104004      HLT+4                    ;ANS1 NOT EQUAL TO 052525

014416 022767 052525 164360  CMP      #052525,ANS2   ;DID 052525 GET STORED?
014424 001401      BEQ      .+4            ;BRANCH IF OK
014426 104004      HLT+4                    ;ANS2 NOT EQUAL TO 052525

014430 005767 164352      TST      ANS3            ;DID ZERO GET STORED?
014434 001401      BEQ      .+4            ;BRANCH IF OK
014436 104004      HLT+4                    ;ANS3 NOT EQUAL TO ZERO

014440 005767 164344      TST      ANS4            ;DID ZERO GET STORED?
014444 001401      BEQ      .+4            ;BRANCH IF OK
014446 104004      HLT+4                    ;ANS4 NOT EQUAL TO ZERO

```

```

*****
:TEST 122:      TEST STCDF (STORE-CONVERT DOUBLE TO FLOATING)
:              STORE 125252,125252,125252,125252 --> 125252,125253
:              FPS = 047610, AC = AC3, DST = M2-R1
*****

```

```

014450 104400      SCOPE
014452 000404      BR      TST122

014454 125252 125252 125252 125252  DAT122: 125252,125252,125252,125252
014462 125252

014464 170127 047617      TST122: LDFPS #047617      ;LOAD FLOATING POINT STAUS
014470 005067 164312      CLR      ANS3            ;CLEAR ANS3
014474 005067 164310      CLR      ANS4            ;CLEAR ANS4
014500 172767 177750      LDD      DAT122, AC3     ;LOAD 125252,125252,125252,125252 INTO AC3
014504 012701 001002      MOV      #ANS1, R1
014510 176321      STCDF   AC3, (R1)+      ;STORE-CONVERT AC3 IN ANS1, ANS2
014512 170200      STFPS   FPS              ;STORE FLOATING POINT STATUS
014514 022700 047610      CMP      #047610,FPS     ;CHECK FLOATING POINT STATUS
014520 001401      BEQ      .+4            ;BRANCH IF OK
014522 104000      HLT                               ;FPS NOT EQUAL TO 047610

014524 022701 001006      CMP      #ANS3, R1      ;CHECK R1
014530 001401      BEQ      .+4            ;BRANCH IF OK
014532 104000      HLT                               ;R1 NOT EQUAL TO #ANS3

014534 022767 125252 164240  CMP      #125252,ANS1   ;DID 125252 GET STORED?
014542 001401      BEQ      .+4            ;BRANCH IF OK
014544 104004      HLT+4                    ;ANS1 NOT EQUAL TO 125252

```

```

014546 022767 125253 164230      CMP      #125253,ANS2      :DID 125253 GET STORED?
014554 001401      BEQ      .+4              :BRANCH IF OK
014556 104004      HLT+4                    :ANS2 NOT EQUAL TO 125253

014560 005767 164222      TST      ANS3              :DID ZERO GET STORED?
014564 001401      BEQ      .+4              :BRANCH IF OK
014566 104004      HLT+4                    :ANS3 NOT EQUAL TO ZERO

014570 005767 164214      TST      ANS4              :DID ZERO GET STORED?
014574 001401      BEQ      .+4              :BRANCH IF OK
014576 104004      HLT+4                    :ANS4 NOT EQUAL TO ZERO

```

```

:*****
:TEST 123:      TEST STCDF (STORE-CONVERT DOUBLE TO FLOATING)
:      STORE 140125,052525,052525,052525 --> 140125,052525
:      FPS = 047610, AC = AC2, DST = M3-R5
:*****

```

```

014600 104400      SCOPE
014602 000405      BR      TST123

014604 140125 052525 052525 DAT123: 140125,052525,052525,052525
014612 052525      ADR123: ANS1
014614 001002

014616 170127 047617      TST123: LDFPS #047617      :LOAD FLOATING POINT STATUS
014622 005067 164160      CLR      ANS3              :CLEAR ANS3
014626 005067 164156      CLR      ANS4              :CLEAR ANS4
014632 012705 014614      MOV      #ADR123, R5       :LOAD R5
014636 172667 177742      LDD      DAT123, AC2       :LOAD 140125,052525,052525,052525 INTO AC2
014642 176235      STCDF   AC2, 2(R5)+        :STORE-CONVERT AC2 IN ANS1, ANS2
014644 170200      STFPS   FPS                :STORE FLOATING POINT STATUS
014646 022700 047610      CMP      #047610,FPS       :CHECK FLOATING POINT STATUS
014652 001401      BEQ      .+4              :BRANCH IF OK
014654 104000      HLT                    :FPS NOT EQUAL TO 047610

014656 022705 014616      CMP      #TST123, R5       :CHECK R5
014662 001401      BEQ      .+4              :BRANCH IF OK
014664 104000      HLT                    :R5 NOT EQUAL TO #TST123

014666 022767 140125 164106      CMP      #140125,ANS1      :DID 140125 GET STORED?
014674 001401      BEQ      .+4              :BRANCH IF OK
014676 104004      HLT+4                    :ANS1 NOT EQUAL TO 140125

014700 022767 052525 164076      CMP      #052525,ANS2      :DID 052525 GET STORED?
014706 001401      BEQ      .+4              :BRANCH IF OK
014710 104004      HLT+4                    :ANS2 NOT EQUAL TO 052525

014712 005767 164070      TST      ANS3              :DID ZERO GET STORED?
014716 001401      BEQ      .+4              :BRANCH IF OK
014720 104004      HLT+4                    :ANS3 NOT EQUAL TO ZERO

014722 005767 164062      TST      ANS4              :DID ZERO GET STORED?
014726 001401      BEQ      .+4              :BRANCH IF OK
014730 104004      HLT+4                    :ANS4 NOT EQUAL TO ZERO

```

:TEST 124: TEST STCDF (STORE-CONVERT DOUBLE TO FLOATING)
:STORE 040125,052525,052525,052525 --> 040125,052525
:FPS = 047600, AC = AC1, DST = M4-R3
:*****

014732	104400				SCOPE			
014734	000404				BR	TST124		
014736	040125	052525	052525		DAT124:	040125,052525,052525,052525		
014744	052525							
014746	170127	047617			TST124:	LDFPS #047617	:LOAD FLOATING POINT STATUS	
014752	005067	164030				CLR ANS3	:CLEAR ANS3	
014756	005067	164026				CLR ANS4	:CLEAR ANS4	
014762	012703	001006				MOV #ANS3, R3	:LOAD R3	
014766	172567	177744				LDD DAT124, AC1	:LOAD 040125,052525,052525,052525 INTO AC1	
014772	176143					STCDF AC1, -(R3)	:STORE-CONVERT AC1 IN ANS1, ANS2	
014774	170200					STFPS FPS	:STORE FLOATING POINT STATUS	
014776	022700	047600				CMP #047600, FPS	:CHECK FLOATING POINT STATUS	
015002	001401					BEQ .+4	:BRANCH IF OK	
015004	104000					HLT	:FPS NOT EQUAL TO 047600	
015006	022703	001002				CMP #ANS1, R3	:CHECK R3	
015012	001401					BEQ .+4	:BRANCH IF OK	
015014	104000					HLT	:R3 NOT EQUAL TO #ANS1	
015016	022767	040125	163756			CMP #040125, ANS1	:DID 040125 GET STORED?	
015024	001401					BEQ .+4	:BRANCH IF OK	
015026	104004					HLT+4	:ANS1 NOT EQUAL TO 040125	
015030	022767	052525	163746			CMP #052525, ANS2	:DID 052525 GET STORED?	
015036	001401					BEQ .+4	:BRANCH IF OK	
015040	104004					HLT+4	:ANS2 NOT EQUAL TO 052525	
015042	005767	163740				TST ANS3	:DID ZERO GET STORED?	
015046	001401					BEQ .+4	:BRANCH IF OK	
015050	104004					HLT+4	:ANS3 NOT EQUAL TO ZERO	
015052	005767	163732				TST ANS4	:DID ZERO GET STORED?	
015056	001401					BEQ .+4	:BRANCH IF OK	
015060	104004					HLT+4	:ANS4 NOT EQUAL TO ZERO	

:TEST 125: TEST STCDF (STORE-CONVERT DOUBLE TO FLOATING)
:STORE 140252,125252,125252,125252 --> 140252,125252
:FPS = 047610, AC = AC0, DST = M5-R4
:*****

015062	104400				SCOPE			
015064	000405				BR	TST125		
015066	140252	125252	125252		DAT125:	140252,125252,125252,125252		

```

015074 125252
015076 001002          ADR125: ANS1

015100 170127 047617    TST125: LDFPS #047617      :LOAD FLOATING POINT STATUS
015104 005067 163676      CLR ANS3          :CLEAR ANS3
015110 005067 163674      CLR ANS4          :CLEAR ANS4
015114 012704 015100      MOV #TST125, R4   :LOAD R4
015120 172467 177742      LDD DAT125, ACO   :LOAD 140252,125252,125252,125252 INTO ACO
015124 176054      STCDF ACO, 2-(R4) :STORE-CONVERT ACO IN ANS1, ANS2
015126 170200      STFPS           :STORE FLOATING POINT STATUS
015130 022700 047610      CMP #047610, FPS  :CHECK FLOATING POINT STATUS
015134 001401      BEQ .+4          :BRANCH IF OK
015136 104000      HLT             :FPS NOT EQUAL TO 047610

015140 022704 015076      CMP #ADR125, R4   :CHECK R4
015144 001401      BEQ .+4          :BRANCH IF OK
015146 104000      HLT             :R4 NOT EQUAL TO #ADR125

015150 022767 140252 163624  CMP #140252, ANS1 :DID 140252 GET STORED?
015156 001401      BEQ .+4          :BRANCH IF OK
015160 104004      HLT+4          :ANS1 NOT EQUAL TO 140252

015162 022767 125253 163614  CMP #125253, ANS2 :DID 125253 GET STORED?
015170 001401      BEQ .+4          :BRANCH IF OK
015172 104004      HLT+4          :ANS2 NOT EQUAL TO 125253

015174 005767 163606      TST ANS3          :DID ZERO GET STORED?
015200 001401      BEQ .+4          :BRANCH IF OK
015202 104004      HLT+4          :ANS3 NOT EQUAL TO ZERO

015204 005767 163600      TST ANS4          :DID ZERO GET STORED?
015210 001401      BEQ .+4          :BRANCH IF OK
015212 104004      HLT+4          :ANS4 NOT EQUAL TO ZERO

```

```

*****
:TEST 126: TEST STCDF (STORE-CONVERT DOUBLE TO FLOATING)
: STORE 052525,052525,052525,052525 --> 052525,052525
: FPS = 047600, AC = AC1, DST = M6-R4
*****

```

```

015214 104400          SCOPE
015216 000404          BR TST126

015220 052525 052525 052525 052525  DAT126: 052525,052525,052525,052525
015226 052525

015230 170127 047617    TST126: LDFPS #047617      :LOAD FLOATING POINT STATUS
015234 005067 163546      CLR ANS3          :CLEAR ANS3
015240 005067 163544      CLR ANS4          :CLEAR ANS4
015244 012704 001224      MOV #ANS1+222, R4 :LOAD R4
015250 172567 177744      LDD DAT126, AC1   :LOAD 052525,052525,052525,052525 INTO AC1
015254 176164      STCDF AC1, -222(R4) :STORE-CONVERT AC1 IN ANS1, ANS2
015260 170200      STFPS           :STORE FLOATING POINT STATUS
015262 022700 047600      CMP #047600, FPS  :CHECK FLOATING POINT STATUS
015266 001401      BEQ .+4          :BRANCH IF OK

```

```

015270 104000          HLT          ;FPS NOT EQUAL TO 047600
015272 022704 001224  CMP          #ANS1+222,R4  ;CHECK R4
015276 001401          BEQ          .+4          ;BRANCH IF OK
015300 104000          HLT          ;R4 NOT EQUAL TO #ANS1+222
015302 022767 052525 163472  CMP          #052525,ANS1 ;DID 052525 GET STORED?
015310 001401          BEQ          .+4          ;BRANCH IF OK
015312 104004          HLT+4        ;ANS1 NOT EQUAL TO 052525
015314 022767 052525 163462  CMP          #052525,ANS2 ;DID 052525 GET STORED?
015322 001401          BEQ          .+4          ;BRANCH IF OK
015324 104004          HLT+4        ;ANS2 NOT EQUAL TO 052525
015326 005767 163454          TST          ANS3          ;DID ZERO GET STORED?
015332 001401          BEQ          .+4          ;BRANCH IF OK
015334 104004          HLT+4        ;ANS3 NOT EQUAL TO ZERO
015336 005767 163446          TST          ANS4          ;DID ZERO GET STORED?
015342 001401          BEQ          .+4          ;BRANCH IF OK
015344 104004          HLT+4        ;ANS4 NOT EQUAL TO ZERO

```

```

:*****
:TEST 127: TEST STCDF (STORE-CONVERT DOUBLE TO FLOATING)
: STORE 125252,125252,125252,125252 --> 125252,125253
: FPS = 047610, AC = AC2, DST = M7-R1
:*****

```

```

015346 104400          SCOPE
015350 000405          BR          TST127
015352 125252 125252 125252 125252  DAT127: 125252,125252,125252,125252
015360 125252
015362 001002          ADR127: ANS1
015364 170127 047617  TST127: LDFPS          #047617          ;LOAD FLOATING POINT STATUS
015370 005067 163412          CLR          ANS3          ;CLEAR ANS3
015374 005067 163410          CLR          ANS4          ;CLEAR ANS4
015400 172667 177746          LDD          DAT127,AC2    ;LOAD 125252,125252,125252,125252 INTO AC2
015404 012701 015410          MOV          #ADR127+26,R1 ;LOAD R1
015410 176271 177752          STCDF        AC2, 0-26(R1) ;STORE-CONVERT AC2 IN ANS1, ANS2
015414 170200          STFPS          ;STORE FLOATING POINT STATUS
015416 022700 047610          CMP          #047610,FPS   ;CHECK FLOATING POINT STATUS
015422 001401          BEQ          .+4          ;BRANCH IF OK
015424 104000          HLT          ;FPS NOT EQUAL TO 047610
015426 022701 015410          CMP          #ADR127+26,R1 ;CHECK R1
015432 001401          BEQ          .+4          ;BRANCH IF OK
015434 104000          HLT          ;R1 NOT EQUAL TO #ADR127+26
015436 022767 125252 163336  CMP          #125252,ANS1 ;DID 125252 GET STORED?
015444 001401          BEQ          .+4          ;BRANCH IF OK
015446 104004          HLT+4        ;ANS1 NOT EQUAL TO 125252
015450 022767 125253 163326  CMP          #125253,ANS2 ;DID 125253 GET STORED?

```

```

015456 001401      BEQ      .+4      ;BRANCH IF OK
015460 104004      HLT+4          ;ANS2 NOT EQUAL TO 125253

015462 005767 163320 TST      ANS3      ;DID ZERO GET STORED?
015466 001401      BEQ      .+4      ;BRANCH IF OK
015470 104004      HLT+4          ;ANS3 NOT EQUAL TO ZERO

015472 005767 163312 TST      ANS4      ;DID ZERO GET STORED?
015476 001401      BEQ      .+4      ;BRANCH IF OK
015500 104004      HLT+4          ;ANS4 NOT EQUAL TO ZERO

```

```

:*****
:TEST 130:      TEST STCDF (STORE-CONVERT DOUBLE TO FLOATING)
:      STORE 140125,152525,052525,052525 --> 140125
:      FPS = 047610, AC = AC3, DST = M2-R7
:*****

```

```

015502 104400      SCOPE
015504 000404      BR      TST130

015506 140125 152525 052525 DAT130: 140125,152525,052525,052525
015514 052525

015516 170127 047617 TST130: LDFPS #047617 ;LOAD FLOATING POINT STATUS
015522 005067 163260 CLR      ANS3      ;CLEAR ANS3
015526 005067 163256 CLR      ANS4      ;CLEAR ANS4
015532 172767 177750 LDD      DAT130, AC3 ;LOAD 140125,152525,052525,052525 INTO AC3
015536 176327 STCDF   AC3, (PC)+ ;STORE-CONVERT AC3 IN NEXT LOCATION
015540 000000 ADR130: 0 ;LOCATION FOR ANSWER
015542 000402 BR      .+6
015544 000000 HALT
015546 000000 HALT ;PC CATCHER
015550 016767 177764 163224 MOV      ADR130, ANS1
015556 016767 177760 163220 MOV      ADR130+2, ANS2
015564 016767 177754 163214 MOV      ADR130+4, ANS3
015572 016767 177750 163210 MOV      ADR130+6, ANS4
015600 170200 STFPS   FPS ;STORE FLOATING POINT STATUS
015602 022700 047610 CMP      #047610, FPS ;CHECK FLOATING POINT STATUS
015606 001401 BEQ      .+4 ;BRANCH IF OK
015610 104000 HLT ;FPS NOT EQUAL TO 047610

015612 022767 140125 163162 CMP      #140125, ANS1 ;DID 140125 GET STORED?
015620 001401 BEQ      .+4 ;BRANCH IF OK
015622 104004 HLT+4 ;ANS1 NOT EQUAL TO 140125

015624 022767 000402 163152 CMP      #402, ANS2 ;DID "BR .+6" GET STORED?
015632 001401 BEQ      .+4 ;BRANCH IF OK
015634 104004 HLT+4 ;ANS2 NOT EQUAL TO F"BR .+6"

015636 005767 163144 TST      ANS3      ;DID ZERO GET STORED?
015642 001401 BEQ      .+4 ;BRANCH IF OK
015644 104004 HLT+4 ;ANS3 NOT EQUAL TO ZERO

015646 005767 163136 TST      ANS4      ;DID ZERO GET STORED?
015652 001401 BEQ      .+4 ;BRANCH IF OK

```

015654 104004

HLT+4

;ANS4 NOT EQUAL TO ZERO

```

:*****
:TEST 131:      TEST STCDF (STORE-CONVERT DOUBLE TO FLOATING)
:      STORE   040125,052525,052525,052525 --> 040125,052525
:      FPS = 047600, AC = AC1, DST = M3-R7
:*****

```

015656 104400
015660 000404

SCOPE
BR TST131

015662 040125 052525 052525 DAT131: 040125,052525,052525,052525
015670 052525

```

015672 170127 047617 TST131: LDFPS #047617 ;LOAD FLOATING POINT STAUS
015676 005067 163104 CLR ANS3 ;CLEAR ANS3
015702 005067 163102 CLR ANS4 ;CLEAR ANS4
015706 172567 177750 LDD DAT131, AC1 ;LOAD 040125,052525,052525,052525 INTO AC1
015712 176137 001002 STCDF AC1, @#ANS1 ;STORE-CONVERT AC1 IN ANS1, ANS2
015716 170200 STFPS FPS ;STORE FLOATING POINT STATUS
015720 022700 047600 CMP #047600,FPS ;CHECK FLOATING POINT STATUS
015724 001401 BEQ .+4 ;BRANCH IF OK
015726 104000 HLT ;FPS NOT EQUAL TO 047600

```

```

015730 022767 040125 163044 CMP #040125,ANS1 ;DID 040125 GET STORED?
015736 001401 BEQ .+4 ;BRANCH IF OK
015740 104004 HLT+4 ;ANS1 NOT EQUAL TO 040125

```

```

015742 022767 052525 163034 CMP #052525,ANS2 ;DID 052525 GET STORED?
015750 001401 BEQ .+4 ;BRANCH IF OK
015752 104004 HLT+4 ;ANS2 NOT EQUAL TO 052525

```

```

015754 005767 163026 TST ANS3 ;DID ZERO GET STORED?
015760 001401 BEQ .+4 ;BRANCH IF OK
015762 104004 HLT+4 ;ANS3 NOT EQUAL TO ZERO

```

```

015764 005767 163020 TST ANS4 ;DID ZERO GET STORED?
015770 001401 BEQ .+4 ;BRANCH IF OK
015772 104004 HLT+4 ;ANS4 NOT EQUAL TO ZERO

```

```

:*****
:TEST 132:      TEST STCDF (STORE-CONVERT DOUBLE TO FLOATING)
:      STORE   140252,125252,125252,125252 --> 140252,125252
:      FPS = 047610, AC = AC2, DST = M7-R7
:*****

```

015774 104400
015776 000405

SCOPE
BR TST132

016000 140252 125252 125252 DAT132: 140252,125252,125252,125252

016006 125252
016010 001002 ADR132: ANS1

016012 170127 047617 TST132: LDFPS #047617 ;LOAD FLOATING POINT STAUS

016016	005067	162764	CLR	ANS3	:CLEAR ANS3
016022	005067	162762	CLR	ANS4	:CLEAR ANS4
016026	172667	177746	LDD	DAT132, AC2	:LOAD 140252,125252,125252,125252 INTO AC2
016032	176277	177752	STCDF	AC2, QADR132	:STORE-CONVERT AC2 IN ANS1, ANS2
016036	170200		STFPS	FPS	:STORE FLOATING POINT STATUS
016040	022700	047610	CMP	#047610.FPS	:CHECK FLOATING POINT STATUS
016044	001401		BEQ	+.4	:BRANCH IF OK
016046	104000		HLT		:FPS NOT EQUAL TO 047610
016050	022767	140252	162724	CMP	#140252,ANS1
016056	001401			BEQ	+.4
016060	104004			HLT+.4	:DID 140252 GET STORED? :BRANCH IF OK :ANS1 NOT EQUAL TO 140252
016062	022767	125253	162714	CMP	#125253,ANS2
016070	001401			BEQ	+.4
016072	104004			HLT+.4	:DID 125253 GET STORED? :BRANCH IF OK :ANS2 NOT EQUAL TO 125253
016074	005767	162706		TST	ANS3
016100	001401			BEQ	+.4
016102	104004			HLT+.4	:DID ZERO GET STORED? :BRANCH IF OK :ANS3 NOT EQUAL TO ZERO
016104	005767	162700		TST	ANS4
016110	001401			BEQ	+.4
016112	104004			HLT+.4	:DID ZERO GET STORED? :BRANCH IF OK :ANS4 NOT EQUAL TO ZERO

016114	104400			DONE:	SCOPE		
016116	032737	002000	177570		BIT	#SW10,2#SWR	:RING THE BELL?
016124	001005				BNE	1\$:NO!
016126	012767	000207	001242		MOV	#BELL,TYPE	:TYPE A BELL
016134	000004	017376			TYPE	..TYPE	
016140	005046			1\$:	CLR	-(6)	:CLEAR TRACE TRAP
016142	032737	010000	177570		BIT	#SW12,2#SWR	:RUN WITH TRT?
016150	001010				BNE	2\$	
016152	005167	001222			COM	TRPB	
016156	100005				BPL	2\$	
016160	052716	000020			BIS	#20,(6)	:SET TRACE TRAP
016164	012746	001062			MOV	#BEGIN,-(6)	:JUMP TO START OF TEST
016170	000412				BR	YESRT	
016172	012746	001062		2\$:	MOV	#BEGIN,-(6)	:JUMP TO START OF TEST
016176	013700	000042			MOV	2#42,RC	:GET MONITOR ADDRESS
016202	001404				BEQ	3\$:IF NONE
016204	004710				JSR	7,(0)	:GO TO MONITOR
016206	000240				NOP		
016210	000240				NOP		
016212	000240				NOP		
016214	000002			3\$:	RTI		
016216	000002			YESRT:	RTI		:RETURN TO PROGRAM FROM TRAP
016220	032737	000400	177570	.EMT:	BIT	#SW08,2#SWR	:KILL LDUB OR LOOP ON SPEC. TEST
016226	001404				BEQ	1\$	
016230	123767	177570	162542		CMPB	2#SWR,ICNT	:ON RIGHT TEST? *SW7-0*
016236	001437				BEQ	OVER	
016240	113703	177570		1\$:	MOVB	2#SWR,R3	:GET UB BITS
016244	170003				LDUB		
016246	032737	040000	177570		BIT	#SW14,2#SWR	:LOOP ON TEST
016254	001026				BNE	KIT	
016256	032737	004000	177570		BIT	#SW11,2#SWR	:KILL ITERATIONS
016264	001012				BNE	SAVLAD	
016266	105767	162507			TSTB	ICNT+1	
016272	001404				BEQ	2\$:BRANCH IF FIRST
016274	126767	001106	162477		CMPB	TIMES,ICNT+1	:DONE?
016302	001013				BNE	KIT	:BRANCH IF NOT
016304	112767	000001	162467	2\$:	MOVB	#1,ICNT+1	:FIRST ITERATION
016312	105267	162462		SAVLAD:	INCB	ICNT	:COUNT TEST NUMBERS
016316	011667	001060			MOV	(6),LAD	:SAVE LOOP ADDRESS
016322	016737	162452	177570		MOV	ICNT,2#DISPLAY	:DISPLAY TEST NO. AND ITERATION COUNT
016330	000002				RTI		:RETURN
016332	105267	162443		KIT:	INCB	ICNT+1	
016336	016737	162436	177570	OVER:	MOV	ICNT,2#DISPLAY	:SET UP DISPLAY
016344	005767	001032			TST	LAD	:FIRST ONE?
016350	001760				BEQ	SAVLAD	
016352	016716	001024			MOV	LAD,(6)	:FUDGE RETURN ADDRESS
016356	000002				RTI		:FIXES PS

016360	032737	002000	177570	.TRP:	BIT	#SW10,@#SWR	:BELL ON ERROR?
016366	001405				BEQ	1\$:NO - SKIP
016370	012767	000207	001000		MOV	#BELL,.TYPE	:TYPE A BELL
016376	000004	017376			TYPE	.TYPE	
016402	004767	000406		1\$:	JSR	PC,ERROR	:COUNT THE NUMBER OF ERRORS
016406	010446				MOV	R4,-(6)	
016410	032737	020000	177570		BIT	#SW13,@#SWR	:SKIP TYPEOUT IF SET
016416	001072				BNE	4\$	
016420	000004	017344			TYPE	RETURN	
016424	016646	000002			MOV	2(6),-(6)	:PUT ADDRESS OF INSTRUCTION ON STACK
016430	162716	000002			SUB	#2,(6)	
016434	011605				MOV	(6),TTY	:TYPE (6) IN OCTAL
016436	004767	000212			JSR	%7,PRINTR	:TYPE LEADING ZERO'S
016442	000004	017352			TYPE	,SPACE+3	
016446	010005				MOV	R0,TTY	:TYPE R0 IN OCTAL
016450	004767	000200			JSR	%7,PRINTR	:TYPE LEADING ZERO'S
016454	000004	017353			TYPE	,SPACE+4	
016460	012703	001002			MOV	#ANS1,R3	:ADDRESS OF DATA
016464	113604				MOVB	2(6)+,R4	:AMOUNT OF DATA IN TABLE
016466	001426				BEQ	3\$	
016470	100016				BPL	2\$:TYPE STACK?
016472	016667	000006	162302		MOV	6(6),ANS1	
016500	016667	000010	162276		MOV	10(6),ANS2	
016506	016667	000012	162272		MOV	12(6),ANS3	
016514	016667	000014	162266		MOV	14(6),ANS4	
016522	042704	177600			BIC	#177600,R4	:CLEAR SIGN
016526	000004	017353		2\$:	TYPE	,SPACE+4	
016532	012305				MOV	(3)+,TTY	:TYPE (3)+ IN OCTAL
016534	004767	000114			JSR	%7,PRINTR	:TYPE LEADING ZERO'S
016540	005304				DEC	R4	
016542	001371				BNE	2\$	
016544	005700			3\$:	TST	FPS	
016546	100016				BPL	4\$	
016550	000004	017347			TYPE	,SPACE	
016554	170367	162242			STST	FEC	
016560	016705	162236			MOV	FEC,TTY	:TYPE FEC IN OCTAL
016564	004767	000064			JSR	%7,PRINTR	:TYPE LEADING ZERO'S
016570	000004	017352			TYPE	,SPACE+3	
016574	016705	162224			MOV	FEA,TTY	:TYPE FEA IN OCTAL
016600	004767	000050			JSR	%7,PRINTR	:TYPE LEADING ZERO'S
016604	012604			4\$:	MOV	(6)+,R4	
016606	005737	177570			TST	@#SWR	:HALT ON ERROR
016612	100001				BPL	.+4	:SKIP IF CONTINUE
016614	000000				HALT		:HALT ON ERROR!
016616	032737	001000	177570		BIT	#SW09,@#SWR	:CHECK FOR INHIBIT LOOP ON ERROR
016624	001001				BNE	.+4	:SKIP IF LOOP ON ERROR
016626	000002				RTI		
016630	105067	162145			CLRB	ICNT+1	
016634	032737	000400	177570		BIT	#SW08,@#SWR	:CHECK FOR LOAD MICROBREAK
016642	001233				BNE	KIT	:BRANCH IF NOT
016644	113703	177570			MOVB	@#SWR,R3	:PUT MICROBREAK ADDRESS IN R3
016650	170003				LDUB		:LOAD MICROBREAK
016652	000627				BR	KIT	:LOOP ON TEST UNTIL NO ERRORS

```

016654 112767 000001 000130 PRINTR: MOVB #1,A4$ ;SET ZERO FILL SWITCH
016662 000402 BR .+6
016664 005067 000122 PRINTS: CLR A4$ ;SUPRESS LEADING ZERO'S
016670 112767 177772 000115 MOVB #-6,A4$+1 ;SET COUNT
016676 010446 MOV R4,-(6) ;SAVE R4
016700 012704 017002 MOV #3$,R4 ;SET POINTER TO FIRST ASCII CHAR.
016704 105014 CLRB (4) ;CLEAR FIRST BYTE
016706 000405 BR 2$ ;ROTATE FIRST BIT
016710 105014 1$: CLRB (4) ;CLEAR BYTE OF CHARACTER
016712 006105 ROL TTY ;ROTATE BIT INTO C
016714 106114 ROLB (4) ;PACK IT
016716 006105 ROL TTY ;ROTATE BIT INTO C
016720 106114 ROLB (4) ;PACK IT
016722 006105 2$: ROL TTY ;ROTATE BIT INTO C
016724 106114 ROLB (4) ;PACK IT
016726 105714 TSTB (4)
016730 001402 BEQ .+6
016732 105267 000054 INCB A4$
016736 105767 000050 TSTB A4$ ;CHECK FILL SWITCH
016742 001402 BEQ .+6
016744 152724 000060 BISB #'0,(4)+ ;MAKE INTO ASCII CHAR
016750 105267 000037 INCB A4$+1
016754 001355 BNE 1$ ;REPEAT
016756 022704 017002 CMP #3$,R4
016762 001002 BNE .+5
016764 112724 000060 MOVB #'0,(4)+
016770 105014 CLRB (4)
016772 000004 017002 TYPE 3$ ;TYPE IT
016776 012604 MOV (6)+,R4 ;RESTORE R4
017000 000207 RTS PC

017002 000004 3$: .BLKW 4
017012 000000 A4$: 0

017014 005267 000364 ERROR: INC ERRORS ;COUNT ERRORS
017020 132737 000001 000041 BITB #1,@#41 ;AUTO MODE?
017026 001412 BEQ 1$ ;NO!
017030 022767 000010 000346 CMP #10,ERRORS ;TOO MANY?
017036 001006 BNE 1$ ;NOT YET
017040 013700 000042 MOV @#42,R0 ;GET ADDRESS
017044 001403 BEQ 1$ ;FORGET IT IF ZERO
017046 005037 000042 CLR @#42 ;ZAP 42
017052 004710 JSR PC,(0) ;CALL THE MONITOR
017054 000207 1$: RTS PC ;RETURN

```

```

017056 012777 017252 000306 POWDOWN: MOV #ILLUP, @UPVEC ;SET FOR FAST UP
017064 012777 000340 000302 MOV #340, @UPVEC+2 ;PRIO:7
017072 170246 STFPS -(6) ;GET THE FPS
017074 170011 SETD ;
017076 174046 STD ACO, -(6) ;SAVE AC'S
017100 174146 STD AC1, -(6)
017102 174246 STD AC2, -(6)
017104 174346 STD AC3, -(6)
017106 172404 LDD AC4, ACO
017110 174046 STD ACO, -(6)
017112 172405 LDD AC5, ACO
017114 174046 STD ACO, -(6)
017116 010046 MOV RO, -(6) ;SAVE REGISTERS
017120 010146 MOV R1, -(6)
017122 010246 MOV R2, -(6)
017124 010346 MOV R3, -(6)
017126 010446 MOV R4, -(6)
017130 010546 MOV R5, -(6)
017132 010667 000220 MOV SP, SAVES ;SAVE SP
017136 012777 017146 000226 MOV #POWUP, @UPVEC ;SET UP VECTOR
017144 000000 HALT

017146 016706 000204 POWUP: MOV SAVE6, SP ;GET SP
017152 005001 CLR R1 ;WAIT LOOP FOR THE TTY
017154 005201 1$: INC R1
017156 001376 BNE 1$
017160 012605 MOV (6)+, R5 ;GET THE REGISTERS
017162 012604 MOV (6)+, R4
017164 012603 MOV (6)+, R3
017166 012602 MOV (6)+, R2
017170 012601 MOV (6)+, R1
017172 012600 MOV (6)+, R0
017174 170011 SETD
017176 172426 LDD (6)+, ACO ;RESTORE THE AC'S
017200 174005 STD ACO, AC5
017202 172426 LDD (6)+, ACO
017204 174004 STD ACO, AC4
017206 172726 LDD (6)+, AC3
017210 172626 LDD (6)+, AC2
017212 172526 LDD (6)+, AC1
017214 172426 LDD (6)+, ACO
017216 170126 LDFPS (6)+ ;RESTORE FPS
017220 012777 017056 000140 MOV #POWDOWN, @DOWNVEC ;SET UP THE POWER DOWN VECTOR
017226 012777 000340 000134 MOV #340, @DOWNVEC+2
017234 000004 017240 TYPE ..+2 ;.ASCIZ <15><12>"POWER"
017250 000002 RTI

017252 000000 ILLUP: HALT ;THE POWER UP SEQUENCE WAS STARTED
017254 000776 BR .-2 ;BEFORE THE POWER DOWN WAS COMPLETE

```

```

017256 010546          .IOT:  MOV      TTY, -(6)          ;SAVE TTY
017260 017605 000002  MOV      @2(6), TTY      ;GET ADDRESS TO BE TYPED
017264 105715          1$:   TSTB   (TTY)          ;TERMINATOR?
017266 001406          BEQ     2$              ;
017270 112537 177566  MOVB   (TTY)+, @#177566 ;LOAD AND TYPE THE CHARACTER
017274 105737 177564  TSTB   @#177564        ;IS THE PRINTER READY
017300 100375          BPL     -4              ;
017302 000770          BR      1$              ;GET THE NEXT CHARACTER
017304 017646 000002  2$:   MOV      @2(6), -(6)    ;GET ADDRESS TO BE TYPED
017310 062766 000002 000004  ADD     #2, 4(6)        ;ADD 2 TO THE ADDRESS
017316 022666 000002  CMP     (6)+, 2(6)     ;IS IT .+2?
017322 001006          BNE     3$              ;NO
017324 062705 000002  ADD     #2, TTY        ;ADD 2 TO THE ADDRESS
017330 042705 000001  BIC     #1, TTY        ;BACK UP TO AN EVEN BYTE
017334 010566 000002  MOV     TTY, 2(6)     ;RESTORE ADDRESS
017340 012605          3$:   MOV     (6)+, TTY     ;RESTORE TTY
017342 000002          RTI                    ;RETURN

017344 005015 000      RETURN: .ASCIZ <15><12>    ;RETURN AND LINEFEED
017347 015      020012 020040 SPACE: .ASCIZ <15><12>" " ;RETURN AND 3 SPACES
017354 000

017356 017356          .EVEN
017356 000000  SAVE6:  0
017360 172160  FPTADR: 172160      ;FLOATING POINT ADDRESS ON THE 11/20
017362 000244 000246  FPVECT: 244, 246   ;FLOATING POINT VECTOR ADDRESS
017366 000024 000026  DWNVEC: 24, 26    ;POWER DOWN VECTOR ADDRESS
017372 000024 000026  UPVEC:  24, 26    ;POWER UP VECTOR ADDRESS
017376 000000          .TYPE:  0
017400 000000          TRPB:   0
017402 000000          LAD:    0          ;LOOP ADDRESS
017404 000000          ERRORS: 0         ;ERROR COUNT
017406 000377          TIMES: 377      ;ITERATION COUNT
017406 000001          .END

```

N06

MAINDEC-11-DCFPI-B
DCFPI.P11

TEST OF LDCDF, LDCFD, STCFD, STCDF
CROSS REFERENCE TABLE -- USER SYMBOLS

MACY11 27(732) 17-SEP-76 10:46 PAGE 79

AC0	=:000000	391#	465*	466	568*	574	626*	632	657*	658*	673	1000*	1006	1070*
		1077	1297*	1298*	1304	1531*	1532*	1538	1692*	1693	1730*	1736	1900*	1906
		1983*	1989	2094*	2100	2310*	2312	2530*	2537	2568*	2570	2652*	2653	2694*
		2695	2741*	2742	2824*	2825	2989*	2990	3028*	3029	3439*	3440	3480*	3482
		3659*	3660	4052	4056*	4057	4058*	4059	4081*	4082	4083*	4084	4088*	
AC1	=:000001	392#	510*	516	754*	760	784*	799	864*	870	928*	934	1106*	1112
		1165*	1171	1260*	1261*	1267	1494*	1495*	1501	1654*	1655*	1661	1817*	1823
		2021*	2027	2130*	2131	2238*	2239	2420*	2421	2782*	2783	3184*	3186	3263*
		3264	3302*	3303	3342*	3343	3614*	3615	3703*	3704	3838*	3839	4053	4087*
AC2	=:000002	393#	597*	603	696*	702	862*	864	964*	976	1136*	1142	1371*	1372*
		1378	1408*	1409*	1415	1571*	1572*	1587	1617*	1618*	1624	1693*	1699	1773*
		1785	1858*	1864	1941*	1947	2166*	2167	2347*	2349	2384*	2385	2529*	2530
		2908*	2909	3067*	3068	3106*	3107	3391*	3392	3570*	3571	3747*	3749	3878*
		3879	4054	4086*										
AC3	=:000003	394#	481*	487	539*	545	725*	731	823*	838	894*	900	1035*	1041
		1195*	1201	1223*	1224*	1230	1334*	1335*	1341	1448*	1449*	1464	2057*	2063
		2202*	2203	2274*	2275	2456*	2458	2493*	2494	2609*	2611	2864*	2865	2950*
		2951	3145*	3146	3224*	3225	3440*	3446	3524*	3526	3791*	3792	4055	4085*
AC4	=:000004	395#	466*	1223	1260	1297	1334	1371	1408	1494	1531	1617	1654	4056
		4084*												
AC5	=:000005	396#	4058	4082*										
ADR101	011710	2821#	2823	2831										
ADR102	012042	2866#	2870	2871	2872	2873								
ADR104	012272	2947#	2951*											
ADR123	014614	3564#	3569											
ADR125	015076	3653#	3666											
ADR127	015362	3743#	3748	3755										
ADR130	015540	3793#	3797	3798	3799	3800								
ADR132	016010	3873#	3879*											
ADR22	003276	1031#	1042											
ADR24	003512	1103#	1105	1113										
ADR27	003756	1192#	1195											
ADR47	006200	1813#	1816											
ADR51	006432	1896#	1907											
ADR53	006666	1980#	1982	1990										
ADR56	007202	2091#	2094											
ADR75	011174	2648#	2651											
ADR77	011454	2737#	2748											
ANS1	001002	425#	487*	488	516*	517	545*	546	574*	575	603*	604	632*	633
		673#	674	702*	703	731*	732	760*	761	799*	800	838*	839	870*
		871	900*	905	934*	939	968*	976*	977	1006*	1011	1041*	1046	1077*
		1082	1112*	1117	1142*	1143	1171*	1172	1201*	1202	1230*	1231	1267*	1268
		1304*	1305	1341*	1342	1378*	1379	1415*	1416	1464*	1465	1501*	1502	1538*
		1539	1587*	1588	1624*	1625	1661*	1662	1699*	1700	1736*	1741	1777*	1785*
		1786	1823*	1828	1864*	1869	1906*	1911	1947*	1952	1989*	1994	2027*	2028
		2063*	2064	2100*	2101	2131*	2137	2167*	2173	2203*	2209	2239*	2245	2275*
		2281	2312*	2318	2349*	2355	2385*	2391	2421*	2427	2458*	2464	2494*	2500
		2537*	2539	2569	2576	2580	2610	2621	2648	2663	2701*	2710	2737	2752
		2781	2789	2793	2821	2835	2870*	2879	2909*	2915	2947	2957	2990*	2996
		3029*	3035	3068*	3074	3107*	3113	3146*	3152	3186*	3192	3225*	3231	3264*
		3270	3303*	3309	3343*	3358	3392*	3407	3446*	3447	3481	3488	3492	3525
		3536	3564	3581	3621	3625	3653	3670	3702	3710	3714	3743	3759	3797*
		3806	3839*	3845	3873	3885	3968	3972*						
ANS2	001004	426#	492	521	550	579	609	637	678	707	736	765	804	843
		875	909	943	981	1015	1050	1086	1121	1147	1176	1206	1235	1272
		1309	1346	1383	1420	1469	1506	1543	1592	1629	1666	1704	1745	1790

MAINDEC-11-DCFPI-B TEST OF LDCDF, LDCFD, STCFD, STCDF
 DCFPI.P11 CROSS REFERENCE TABLE --- USER SYMBOLS

DATE	TIME	LDCDF	LDCFD	STCFD	STCDF	USER SYMBOLS
01	15220	3696	3703			
01	15352	3740	3747			
00	23220	780	784			
01	15506	3785	3791			
01	15662	3832	3838			
01	16000	3871	3879			
00	24326	819	823			
00	25554	857	862			
00	26550	889	893			901
00	27500	923	927			
00	1312	506	510			
00	3050	957	963			
00	3166	995	1007			
00	3266	1029	1031			1078
00	3370	1064	1069			
00	3502	1100	1103			
00	3660	1161	1165			
00	3746	1190	1192			
00	1400	535	539			
00	4036	1220	1224			
00	4136	1257	1261			
00	4242	1294	1298			
00	4346	1331	1335			
00	4452	1368	1372			
00	4556	1405	1409			
00	4666	1445	1449			
00	5030	1491	1495			
00	1466	564	568			
00	5134	1528	1532			
00	5250	1568	1572			
00	5412	1614	1618			
00	5516	1651	1655			
00	5622	1689	1692			
00	5726	1726	1729			1737
00	6174	1812	1813			
00	1554	593	597			
00	6312	1854	1865			
00	6426	1895	1896			
00	6544	1937	1940			1948
00	6662	1978	1980			
00	7074	2054	2057			
00	7176	2090	2091			
00	7302	2127	2130			
00	1642	622	626			
00	7404	2163	2166			
00	7506	2199	2202			
00	7610	2235	2238			
00	7712	2271	2274			
01	0014	2307	2310			
01	0116	2344	2347			
01	0220	2381	2384			
01	0322	2417	2420			
00	1724	653	658			
01	0424	2453	2456			
01	0526	2490	2493			
01	0630	2526	2529			

551	571	576	580	600	605	609	629	634	638	662	666	670
675	679	699	704	708	728	733	737	757	762	766	788	792
796	801	805	827	831	835	840	844	867	872	876	897	902
906	910	931	936	940	944	973	978	982	1003	1008	1012	1016
1038	1043	1047	1051	1074	1079	1083	1087	1109	1114	1118	1122	1133
1144	1148	1168	1173	1177	1198	1203	1207	1227	1232	1236	1240	1244
1264	1269	1273	1277	1281	1301	1306	1310	1314	1318	1338	1343	1347
1351	1355	1375	1380	1384	1388	1392	1412	1417	1421	1425	1429	1453
1457	1461	1466	1470	1474	1478	1498	1503	1507	1511	1515	1535	1540
1544	1548	1552	1576	1580	1584	1589	1593	1597	1601	1621	1626	1630
1634	1638	1658	1663	1667	1671	1675	1696	1701	1705	1709	1713	1723
1738	1742	1746	1750	1754	1782	1787	1791	1795	1799	1820	1825	1829
1833	1837	1841	1861	1866	1870	1874	1878	1882	1903	1908	1912	1916
1920	1924	1944	1949	1953	1957	1961	1965	1986	1991	1995	1999	2003
2007	2024	2029	2033	2037	2041	2060	2065	2069	2073	2077	2097	2102
2106	2110	2114	2134	2138	2142	2146	2150	2170	2174	2178	2182	2186
2206	2210	2214	2218	2222	2242	2246	2250	2254	2258	2278	2282	2286
2250	2294	2315	2319	2323	2327	2331	2352	2356	2360	2364	2368	2388
2392	2396	2400	2404	2424	2428	2432	2436	2440	2461	2465	2469	2473
2477	2497	2501	2505	2509	2513	2533	2540	2544	2548	2552	2573	2577
2581	2585	2589	2593	2614	2618	2622	2626	2630	2634	2656	2660	2664
2668	2672	2676	2698	2711	2715	2719	2723	2745	2749	2753	2757	2761
2765	2786	2790	2794	2798	2802	2806	2828	2832	2836	2840	2844	2848
2867	2876	2880	2884	2888	2892	2912	2916	2920	2924	2928	2954	2958
2962	2966	2970	2993	2997	3001	3005	3009	3032	3036	3040	3044	3048
3071	3075	3079	3083	3087	3110	3114	3118	3122	3126	3149	3153	3157
3161	3165	3189	3193	3197	3201	3205	3228	3232	3236	3240	3244	3248
3271	3275	3279	3283	3306	3310	3314	3318	3322	3347	3351	3355	3359
3363	3367	3371	3396	3400	3404	3408	3412	3416	3420	3443	3447	3451
3456	3460	3485	3489	3493	3497	3501	3505	3529	3533	3537	3541	3545
3549	3574	3578	3582	3586	3590	3594	3618	3622	3626	3630	3634	3638
3663	3667	3671	3675	3679	3683	3707	3711	3715	3719	3723	3727	3731
3756	3760	3764	3768	3772	3794	3803	3807	3811	3815	3819	3823	3827
3850	3854	3858	3882	3886	3890	3894	3898	3932	3936	3940	3944	3948
4028	4032*	4092	4096	4103	4118*					4005	4009	4013
404	3925**									4015	4019	4023
454	4097**									4021	4025	4029
450	3951**									4027	4031	4035
4520	3952*		3954	4124*						4033	4037	4041
3950*	3906									4039	4043	4047

.
Y B
N

0
017
016
015
014
013
012
011
010
009
008
007
006
005
004
003
002
001

K07

MAINDEC-11-DCFPI-B TEST OF LDCDF, LDCFD, STCFD, STCDF
DCFPI.P11 CROSS REFERENCE TABLE -- MACRO NAMES

MACY11 27(732) 17-SEP-76 10:46 PAGE 90

SSCF67	460#	2297	2334	2443							
SSCF7	460#	2609									
SSCF77	460#	2931									
STCDF	460#	2973	3012	3051	3090	3129	3208	3247	3286	3325	3374
STCFD	460#	2117	2153	2199	2225	2261	2371	2407	2480		

MAINDEC-11-DCFPI-B TEST OF LDCDF, LDCFD, STCFD, STCDF
 DCFPI.P11 CROSS REFERENCE TABLE -- PERMANENT SYMBOLS

	1167	1172	1176	1197	1202	1206	1226	1263	1268	1272	1300	1305	1309	1337	1342
	1346	1374	1379	1383	1411	1452	1456	1460	1465	1469	1473	1477	1497	1502	1506
	1534	1539	1543	1575	1579	1583	1588	1592	1596	1600	1620	1625	1629	1657	1662
	1666	1695	1700	1704	1732	1737	1741	1745	1775	1781	1786	1790	1819	1824	1828
	1832	1860	1865	1869	1873	1902	1907	1911	1915	1943	1948	1952	1956	1985	1990
	1994	1998	2023	2028	2059	2064	2068	2096	2101	2105	2133	2137	2141	2169	2173
	2177	2205	2209	2213	2241	2245	2249	2277	2281	2285	2314	2351	2387	2391	2395
	2423	2427	2431	2460	2496	2500	2504	2532	2539	2543	2572	2576	2580	2584	2613
	2617	2621	2625	2655	2659	2663	2667	2697	2705	2710	2714	2744	2748	2752	2756
	2785	2789	2793	2797	2827	2831	2835	2839	2875	2879	2883	2911	2915	2919	2953
	2957	2961	2992	2996	3000	3031	3035	3039	3070	3074	3078	3109	3113	3117	3148
	3152	3156	3188	3227	3231	3235	3266	3270	3274	3305	3309	3313	3346	3350	3354
	3358	3362	3395	3399	3403	3407	3411	3442	3447	3451	3484	3488	3492	3496	3528
	3532	3536	3540	3573	3577	3581	3585	3617	3621	3625	3629	3662	3666	3670	3674
	3706	3710	3714	3718	3751	3755	3759	3763	3802	3806	3810	3841	3845	3849	3881
	3885	3889	4027	4041	4107										
CMPB	3927	3937													
COM	3910														
DEC	3980														
EMT	378														
HALT	412	420	2868	2869	3795	3796	3994	4068	4095						
INC	4038	4072													
INCB	3940	3945	4021	4025											
IOT	379														
JMP	415														
JSR	3918	3955	3963	3966	3979	3987	3990	4046							
LDCDF	481	510	539	568	597	626	658	696	725	754	784	823	864	894	928
	964	1000	1035	1070	1106	1136	1165	1195							
LDCFD	1224	1261	1298	1335	1372	1409	1449	1495	1532	1572	1618	1655	1693	1730	1773
	1817	1858	1900	1941	1983	2021	2057	2094							
LDD	465	862	1223	1260	1297	1334	1371	1408	1448	1494	1531	1571	1617	1654	1692
	2989	3028	3067	3106	3145	3184	3224	3263	3302	3342	3391	3439	3480	3524	3570
LDF	3614	3659	3703	3747	3791	3838	3878	4056	4058	4081	4083	4085	4086	4087	4088
	657	2130	2166	2202	2238	2274	2310	2347	2384	2420	2456	2493	2529	2568	2609
LDFPS	2652	2694	2741	2782	2824	2864	2908	2950							
	464	480	509	538	567	596	625	656	695	724	753	783	822	860	892
	926	961	998	1033	1068	1104	1135	1164	1194	1222	1259	1296	1333	1370	1407
	1447	1493	1530	1570	1616	1653	1691	1728	1769	1815	1856	1898	1939	1981	2020
	2056	2093	2129	2165	2201	2237	2273	2309	2311	2346	2348	2383	2419	2455	2457
	2492	2528	2567	2608	2650	2692	2739	2780	2822	2863	2907	2949	2986	3025	3064
	3103	3142	3181	3185	3221	3260	3299	3339	3388	3436	3477	3521	3566	3610	3655
LDUB	3699	3744	3788	3835	3875	4089									
MOV	3930	4002													
	436	437	439	442	445	446	447	448	449	450	451	452	453	454	455
	456	457	893	927	962	963	968	969	999	1034	1069	1071	1105	1729	1770
	1771	1772	1777	1778	1816	1857	1899	1940	1982	2569	2610	2651	2693	2701	2702
	2703	2704	2707	2740	2781	2823	2870	2871	2872	2873	3481	3525	3569	3613	3658
	3702	3748	3797	3798	3799	3800	3905	3913	3915	3916	3941	3942	3946	3949	3953
	3956	3960	3962	3965	3968	3972	3973	3974	3975	3978	3986	3989	3991	4008	4009
	4032	4043	4048	4049	4060	4061	4062	4063	4064	4065	4066	4067	4070	4074	4075
	4076	4077	4078	4079	4090	4091	4097	4098	4105	4111	4112				
MOVE	3929	3939	3969	4001	4004	4007	4029	4101							
NOP	3919	3920	3921												
ROL	4013	4015	4017												
ROLB	4014	4016	4018												
RTI	421	3922	3923	3943	3950	3997	4093	4113							

MAINDEC-11-DCFFI-8 TEST OF LDCFD, LDCFD, STCFD, STCFD
DCFFI.P11 CROSS REFERENCE TABLE -- PERMANENT SYMBOLS

.IFED	477	481	506	510	535	539	564	568	593	597	622	626	652	657	692
	721	725	750	754	780	784	819	823	1213	1231	1251	1276	1288	1313	
.IFNE	1350	1362	1387	1398	1416	1485	1510	1522	1547	1608	1633	1645	1670	1687	
	483	501	512	530	541	559	570	588	599	617	628	646	660	687	
	716	727	745	756	774	786	813	825	866	896	930	1002	1037	1073	
	1138	1167	1197	1213	1226	1231	1250	1263	1268	1287	1300	1305	1324	1337	
	1361	1374	1379	1398	1411	1416	1437	1451	1484	1497	1502	1521	1534	1559	
	1574	1607	1620	1625	1644	1657	1662	1695	1732	1819	1860	1902	1943	1988	
	2059	2096	2122	2133	2158	2169	2194	2205	2230	2241	2266	2277	2314	2337	
	2387	2412	2423	2460	2485	2496	2532	2572	2613	2655	2697	2744	2785	2826	
	2911	2953	2978	2992	3017	3031	3056	3070	3095	3109	3134	3148	3188	3213	
	3252	3266	3291	3305	3330	3345	3379	3394	3442	3484	3528	3573	3617	3662	
.IFN2	487	516	545	574	603	632	665	702	731	760	791	795	820	834	
	900	934	1006	1041	1077	1112	1142	1171	1201	1230	1267	1304	1341	1374	
	1415	1456	1460	1501	1538	1579	1624	1661	1699	1736	1823	1864	1906	1947	
	2027	2063	2100	2137	2173	2209	2245	2281	2318	2355	2391	2427	2464	2500	
	2576	2617	2659	2701	2748	2789	2831	2879	2915	2957	2996	3035	3074	3113	
	3192	3231	3270	3309	3350	3354	3399	3403	3446	3488	3532	3577	3621	3666	
.LIST	327	372	412	422	496	525	554	583	612	641	682	711	740	769	
	808	847	879	913	947	985	1019	1054	1090	1125	1151	1180	1210	1247	1284
	1321	1358	1395	1432	1481	1518	1555	1604	1641	1678	1716	1757	1802	1844	1884
	1927	1968	2010	2044	2080	2117	2153	2189	2225	2261	2297	2334	2371	2407	2443
	2480	2516	2555	2596	2637	2679	2726	2768	2809	2851	2895	2931	2973	3012	3051
	3090	3129	3168	3208	3247	3286	3325	3374	3423	3463	3508	3552	3597	3641	3686
.MACR	372	460	3822	3861	3901	3951	4004	4048	4093	4097					
.MACRO	372	460													
.NLIST	327	372	412	422	460	496	525	554	583	612	641	682	711	740	769
	808	847	879	913	947	985	1019	1054	1090	1125	1151	1180	1210	1247	1284
	1321	1358	1395	1432	1481	1518	1555	1604	1641	1678	1716	1757	1802	1844	1884
	1927	1968	2010	2044	2080	2117	2153	2189	2225	2261	2297	2334	2371	2407	2443
	2480	2516	2555	2596	2637	2679	2726	2768	2809	2851	2895	2931	2973	3012	3051
	3090	3129	3168	3208	3247	3286	3325	3374	3423	3463	3508	3552	3597	3641	3686
.PAGE	460	3901													
.FORM	372	412	422	460	496	525	554	583	612	641	682	711	740	769	
.TEXT	327	372	422	460	496	525	554	583	612	641	682	711	740	769	
.TITLE	327	372	422	460	496	525	554	583	612	641	682	711	740	769	
.END	327	372	422	460	496	525	554	583	612	641	682	711	740	769	

ERRORS DETECTED: 0
DEFAULT GLOBALS GENERATED: 0

*DCFFI, SEQ/SOL/CRF/PAGNUM=DCFFI
RUN-TIME: 28 40 5 SECONDS
RUN-TIME RATIO: 344/75=4.5
CORE USED: 21K (41 PAGES)

