

# FP11

DKFPS, STEPS, CFC, CSETF/D, SETI/L

## MD-11-DCFPA-B

EP DCFPA B DL A

OCT 1976

COPYRIGHT ©1976

**digital**

FICHE 1 OF 1

Made in U.S.A.

This microfiche card contains a grid of frames. The leftmost column consists of frames with technical data, including labels like 'T.F. No.', 'T.C. No.', and 'T.P. No.', along with various alphanumeric codes and symbols. The remaining columns contain frames with diagrams, including flowcharts and schematic drawings. The diagrams show interconnected components and lines, typical of a system architecture or process flow. The entire card is oriented vertically, with the data and diagrams arranged in a grid pattern.

.REM %

IDENTIFICATION

PRODUCT CODE: MAINDEC-11-DCFPA TO DCFPL-B-D  
 PRODUCT NAME: FP11 BASIC INSTRUCTION TESTS  
 DATE CREATED: 25-APR-72  
 MAINTAINER: DIAGNOSTIC GROUP  
 AUTHORS: BOB BRAIN & KEN CHAPMAN

COPYRIGHT (C) 1972  
 DIGITAL EQUIPMENT CORPORATION  
 MAYNARD, MASSACHUSETTS 01754

MAINDEC NO.	INSTRUCTIONS TESTED
DCFPA-B	LDFPS, STFPS, SETI, SETL SETF, SETD, CFCC
DCFPB-A	STST
DCFPC-A	LDF, LDD, STF, STD
DCFPD-A	ADDF, ADD, SUBF, SUBD
DCFDF-A	CMDF, CMPD
DCFPI-A	MULF, MULD
DCFPG-A	DIVF, DIVD
DCFPH-A	CLRF, CLRD, TSTF, TSTD ABSF, ABSD, NEGF, NEGD
DCFFI-A	LDCF, LCCF, STCF, STCF
DCFPI-A	LDCF, LDCLF, LDCID, LDCLD STCFI, STCFL, STCDI, STCDL
DCFFK-A	LDEXP, STEXP
DCFPL-A	MODF, MODD

11-DCFPA-B  
 11-DCFPB-A  
 11-DCFPC-A  
 11-DCFPD-A  
 11-DCFDF-A  
 11-DCFPI-A  
 11-DCFPG-A  
 11-DCFPH-A  
 11-DCFFI-A  
 11-DCFPI-A  
 11-DCFFK-A  
 11-DCFPL-A



FP11 BASIC INSTRUCTION TEST DCFPA - DCFPL  
DESCRIPTION

PAGE 3

1. ABSTRACT

THESE PROGRAMS TEST THE FP11 IN ALL MODES WITH FIXED NUMBER PATTERNS. THE PROGRAMS SHOULD BE RUN IN ORDER FOR AT LEAST 2 PASSES WITH ALL SWITCHES DOWN.

2. REQUIREMENTS

2.1 EQUIPMENT

PDP11/45 STANDARD COMPUTER WITH FP11 OPTION

2.2 STORAGE

PROGRAM STORAGE - THE ROUTINES USE MEMORY 0 - 17776

2.3 PRELIMINARY PROGRAMS

NONE

3. LOADING PROCEDURE

USE STANDARD PROCEDURE FOR ABS TAPES.

4. STARTING PROCEDURE

4.1 CONTROL SWITCH SETTINGS

SEE 5.1.1 (ALL DOWN FOR WORST CASE TESTING)

4.2 STARTING ADDRESS

THE PROGRAM SHOULD ALWAYS BE STARTED AT 200.

4.3 PROGRAM AND/OR OPERATOR ACTION

- 1) LOAD PROGRAM INTO MEMORY USING ABS LOADER.
- 2) LOAD ADDRESS 200.
- 3) SET SWITCHES (SEE SEC 5.1.1) ALL DOWN FOR WORST CASE
- 4) PRESS START.
- 5) THE PROGRAM WILL LOOP AND BELL WILL RING ONCE EVERY PASS
- 6) A MINIMUM OF TWO PASSES SHOULD ALWAYS BE RUN.

.....

E01

MAINDEC-11-DOFPA-B  
DOFPAE.P11

TEST OF LDFPS, STFPS, SETF, SETD, SETI, SETL, CFCC

MACY11 27(732) 10-SEP-76 11:04 PAGE 4

104  
105

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

DATA DISPLAY SWITCH TO THE DISPLAY POSITION.

5. OPERATING PROCEDURE

5.1 OPERATIONAL SWITCH SETTINGS

AT SA 200 . . . ALL SWITCHES DOWN IS WORST CASE TESTING. IF AN ERROR OCCURS, THAT TEST WILL BE LOOPED UPON UNTIL COMPLETION OF 256 CONSECUTIVE PASSES WITH NO ERRORS OF THE SUBTEST IF SW<9> SET TO A 1. THE BELL WILL RING UPON COMPLETION OF A PASS.

5.1.1 SWITCH SETTINGS ARE:

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

5.2 SUBROUTINE ABSTRACTS

5.2.1 SCOPE

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

5.2.2 HLT

THIS ROUTINE PRINTS OUT AN ERROR MESSAGE (SEE 6.1.) IF A HLT IS EXECUTED. THE SUBTEST WILL BE LOOPED UPON UNTIL 256 CONSECUTIVE GOOD PASSES ARE COMPLETED IF SW<9> IS ON A 1. TO INHIBIT TYPEOUTS, PUT SW<13> ON A 1.

MACY11 27(732) 10-SEP-76 11:04 PAGE 5







318

%

.TITLE MAINDEC-11-DCFPA-B TEST OF LDFPS, STFPS, SETF, SETD, SETI, SETL, CFCC  
: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      A0= %0
000001      A1= %1
000002      A2= %2
000003      A3= %3
000004      A4= %4
000005      A5= %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      STAQ= 170005
170007      STQ0= 170007
170006      MRS= 170006
170004      LDSC= 170004

000000      .= 0 ;TRAP CATCHER FROM 0 - 776
000200      .= 200
000200 000167 000622      JMP BEG
000760 000760      .= 760
000760 170200      FLTERR: STFPS FPS
000762 170367 000034      STST FEC
000766 000000      HALT
000770 000002      RTI

```

001000	001000			=	1000					
001000	000000			ICNT:	0					: ITERATION COUNT - LH TEST NO. - RH
001002	000000			ANS1:	0					: FIRST ANSWER (SEE CODE)
001004	000000			ANS2:	0					
001006	000000			ANS3:	0					
001010	000000			ANS4:	0					
001012	000000			ANS5:	0					
001014	000000			ANS6:	0					
001016	000000			ANS7:	0					
001020	000000			ANS8:	0					
001022	000000			FEC:	0					: FLOATING EXCEPTION CODES
001024	000000			FEA:	0					: FLOATING EXECPTION ADDRESS
001026	012706	000600		BEG:	MOV	#600,SP				: ** STACK AT 600 **
001032	012737	001054	000004		MOV	#M1120, @#4				: FIND OUT WHICH MACHINE THIS IS
001040	005737	177772			TST	@#177772				: IS PIRQ THERE?
001044	012767	000006	010462		MOV	#6, YESRT				: FUDGE IN RTT IF 11/45
001052	000403				BR	BEGIN				
001054	016737	011616	000010	M1120:	MOV	FPTADR, @#10				: LOAD THE ILLEGAL INSTRUCTION VECTOR
										: WITH THE ADDRESS OF THE FPU.
										: THE FPU WILL HANDLE THE BAD OPCODES
001062	012737	000006	000004	BEGIN:	MOV	#6, @#4				: RESET 4
001070	012706	000600			MOV	#600, SP				
001074	012737	011534	000014		MOV	#YESRT, @#14				: SET TRACE TRAP VECTOR
001102	012777	012374	011574		MOV	#POWDWN, @DOWNVEC				
001110	012777	000340	011570		MOV	#340, @DOWNVEC+2				
001116	012737	012574	000020		MOV	#. IOT, @#20				: SET UP VECTOR 20
001124	012700	000030			MOV	#30, R0				: SET R0 TO VECTOR 30
001130	012720	011676			MOV	#. TRP, (0)+				: SET EMT VECTOR
001134	012720	000340			MOV	#340, (0)+				
001140	012720	011536			MOV	#. EMT, (0)+				: SET TRAP VECTOR
001144	012710	000340			MOV	#340, (0)				
001150	012777	000760	011522		MOV	#FLTERR, @FPVECT				: LOAD INTERRUPT VECTOR
001156	012777	000340	011516		MOV	#340, @FPVECT+2				: LOCK UP PROCESSOR
001164	005067	177610			CLR	ICNT				
001170	005067	011524			CLR	LAD				

L01

\*\*\*\*\*  
:TEST 1: TEST LDFPS (LOAD FLOATING POINT STATUS) AND  
: STFPS (STORE FLOATING POINT STATUS).  
: FPS = 000000, SRC/DST = MO-RO  
\*\*\*\*\*

001174 104400  
001176 012700 000000  
001202 170100  
001204 022700 000000  
001210 001402  
001212 005000  
001214 104000

TST1: SCOPE  
MOV #000000,RO ;SET UP RO  
LDFPS RO ;LOAD FLOATING POINT STATUS WITH 000000  
CMP #000000,RO ;CHECK RO  
BEQ TSA1 ;BRANCH IF OK  
CLR FPS ;FPS NOT YET STORED  
HLT ;RO NOT EQUAL TO 000000

001216 170200  
001220 010000  
001222 022700 000000  
001226 001401  
001230 104000

TSA1: STFPS RO ;STORE FLOATING POINT STATUS IN RO  
MOV RO, FPS ;SAVE FPS FOR TYPING  
CMP #000000,RO ;CHECK FLOATING POINT STATUS  
BEQ .+4 ;BRANCH IF OK  
HLT ;FPS NOT EQUAL TO 000000

\*\*\*\*\*  
:TEST 2: TEST LDFPS (LOAD FLOATING POINT STATUS) AND  
: STFPS (STORE FLOATING POINT STATUS).  
: FPS = 147757, SRC/DST = MO-R1  
\*\*\*\*\*

001232 104400  
001234 012701 177757  
001240 170101  
001242 022701 177757  
001246 001402  
001250 005000  
001252 104000

TST2: SCOPE  
MOV #177757,R1 ;SET UP R1  
LDFPS R1 ;LOAD FLOATING POINT STATUS WITH 177757  
CMP #177757,R1 ;CHECK R1  
BEQ TSA2 ;BRANCH IF OK  
CLR FPS ;FPS NOT YET STORED  
HLT ;R1 NOT EQUAL TO 177757

001254 170201  
001256 010100  
001260 022701 147757  
001264 001401  
001266 104000

TSA2: STFPS R1 ;STORE FLOATING POINT STATUS IN R1  
MOV R1, FPS ;SAVE FPS FOR TYPING  
CMP #147757,R1 ;CHECK FLOATING POINT STATUS  
BEQ .+4 ;BRANCH IF OK  
HLT ;FPS NOT EQUAL TO 147757

\*\*\*\*\*  
:TEST 3: TEST LDFPS (LOAD FLOATING POINT STATUS) AND  
: STFPS (STORE FLOATING POINT STATUS).  
: FPS = 000001, SRC/DST = MO-R2  
\*\*\*\*\*

001270 104400  
001272 012702 000001  
001276 170102  
001300 022702 000001  
001304 001402  
001306 005000  
001310 104000

TST3: SCOPE  
MOV #000001,R2 ;SET UP R2  
LDFPS R2 ;LOAD FLOATING POINT STATUS WITH 000001  
CMP #000001,R2 ;CHECK R2  
BEQ TSA3 ;BRANCH IF OK  
CLR FPS ;FPS NOT YET STORED  
HLT ;R2 NOT EQUAL TO 000001

MO1

001312 170202  
001314 010200  
001316 022702 000001  
001322 001401  
001324 104000

TSA3: STFPS R2 ; STORE FLOATING POINT STATUS IN R2  
MOV R2, FPS ; SAVE FPS FOR TYPING  
CMP #000001,R2 ; CHECK FLOATING POINT STATUS  
BEQ .+4 ; BRANCH IF OK  
HLT ; FPS NOT EQUAL TO 000001

\*\*\*\*\*  
:TEST 4: TEST LDFPS (LOAD FLOATING POINT STATUS) AND  
: STFPS (STORE FLOATING POINT STATUS).  
: FPS = 000002, SRC/DST = MO-R3  
\*\*\*\*\*

001326 104400  
001330 012703 000002  
001334 170103  
001336 022703 000002  
001342 001402  
001344 005000  
001346 104000

TST4: SCOPE  
MOV #000002,R3 ; SET UP R3  
LDFPS R3 ; LOAD FLOATING POINT STATUS WITH 000002  
CMP #000002,R3 ; CHECK R3  
BEQ TSA4 ; BRANCH IF OK  
CLR FPS ; FPS NOT YET STORED  
HLT ; R3 NOT EQUAL TO 000002

001350 170203  
001352 010300  
001354 022703 000002  
001360 001401  
001362 104000

TSA4: STFPS R3 ; STORE FLOATING POINT STATUS IN R3  
MOV R3, FPS ; SAVE FPS FOR TYPING  
CMP #000002,R3 ; CHECK FLOATING POINT STATUS  
BEQ .+4 ; BRANCH IF OK  
HLT ; FPS NOT EQUAL TO 000002

\*\*\*\*\*  
:TEST 5: TEST LDFPS (LOAD FLOATING POINT STATUS) AND  
: STFPS (STORE FLOATING POINT STATUS).  
: FPS = 000004, SRC/DST = MO-R4  
\*\*\*\*\*

001364 104400  
001366 012704 000004  
001372 170104  
001374 022704 000004  
001400 001402  
001402 005000  
001404 104000

TST5: SCOPE  
MOV #000004,R4 ; SET UP R4  
LDFPS R4 ; LOAD FLOATING POINT STATUS WITH 000004  
CMP #000004,R4 ; CHECK R4  
BEQ TSA5 ; BRANCH IF OK  
CLR FPS ; FPS NOT YET STORED  
HLT ; R4 NOT EQUAL TO 000004

001406 170204  
001410 010400  
001412 022704 000004  
001416 001401  
001420 104000

TSA5: STFPS R4 ; STORE FLOATING POINT STATUS IN R4  
MOV R4, FPS ; SAVE FPS FOR TYPING  
CMP #000004,R4 ; CHECK FLOATING POINT STATUS  
BEQ .+4 ; BRANCH IF OK  
HLT ; FPS NOT EQUAL TO 000004

\*\*\*\*\*  
:TEST 6: TEST LDFPS (LOAD FLOATING POINT STATUS) AND  
: STFPS (STORE FLOATING POINT STATUS).  
: FPS = 000010, SRC/DST = MO-R5  
\*\*\*\*\*

```

001422 104400
001424 012705 000010   TST6:  SCOPE
001430 170105          MOV      #000010,R5      ;SET UP R5
001432 022705 000010   LDFPS  R5              ;LOAD FLOATING POINT STATUS WITH 000010
001436 001402          CMP      #000010,R5      ;CHECK R5
001440 005000          BEQ     TSA6            ;BRANCH IF OK
001442 104000          CLR     FPS            ;FPS NOT YET STORED
                                HLT     ;R5 NOT EQUAL TO 000010

001444 170205   TSA6:  STFPS  R5              ;STORE FLOATING POINT STATUS IN R5
001446 010500   MOV     R5,      FPS      ;SAVE FPS FOR TYPING
001450 022705 000010   CMP     #000010,R5      ;CHECK FLOATING POINT STATUS
001454 001401   BEQ     .+4            ;BRANCH IF OK
001456 104000   HLT     ;FPS NOT EQUAL TO 000010

```

```

:*****
:TEST 7:          TEST LDFPS (LOAD FLOATING POINT STATUS) AND
:                  STFPS (STORE FLOATING POINT STATUS).
:                  FPS = 000000, SRC/DST = MO-R1
:*****

```

```

001460 104400
001462 012701 000000   TST7:  SCOPE
001466 170101          MOV     #000000,R1      ;SET UP R1
001470 022701 000000   LDFPS  R1              ;LOAD FLOATING POINT STATUS WITH 000000
001474 001402          CMP     #000000,R1      ;CHECK R1
001476 005000          BEQ     TSA7            ;BRANCH IF OK
001500 104000          CLR     FPS            ;FPS NOT YET STORED
                                HLT     ;R1 NOT EQUAL TO 000000

001502 170201   TSA7:  STFPS  R1              ;STORE FLOATING POINT STATUS IN R1
001504 010100   MOV     R1,      FPS      ;SAVE FPS FOR TYPING
001506 022701 000000   CMP     #000000,R1      ;CHECK FLOATING POINT STATUS
001512 001401   BEQ     .+4            ;BRANCH IF OK
001514 104000   HLT     ;FPS NOT EQUAL TO 000000

```

```

:*****
:TEST 10:         TEST LDFPS (LOAD FLOATING POINT STATUS) AND
:                  STFPS (STORE FLOATING POINT STATUS).
:                  FPS = 000040, SRC/DST = MO-R4
:*****

```

```

001516 104400
001520 012704 000040   TST10: SCOPE
001524 170104          MOV     #000040,R4      ;SET UP R4
001526 022704 000040   LDFPS  R4              ;LOAD FLOATING POINT STATUS WITH 000040
001532 001402          CMP     #000040,R4      ;CHECK R4
001534 005000          BEQ     TSA10           ;BRANCH IF OK
001536 104000          CLR     FPS            ;FPS NOT YET STORED
                                HLT     ;R4 NOT EQUAL TO 000040

001540 170204   TSA10: STFPS  R4              ;STORE FLOATING POINT STATUS IN R4
001542 010400   MOV     R4,      FPS      ;SAVE FPS FOR TYPING
001544 022704 000040   CMP     #000040,R4      ;CHECK FLOATING POINT STATUS
001550 001401   BEQ     .+4            ;BRANCH IF OK
001552 104000   HLT     ;FPS NOT EQUAL TO 000040

```

\*\*\*\*\*  
:TEST 11: TEST LDFPS (LOAD FLOATING POINT STATUS) AND  
: STFPS (STORE FLOATING POINT STATUS).  
: FPS = 000100, SRC/DST = MO-R0  
\*\*\*\*\*

001554 104400  
001556 012700 000100  
001562 170100  
001564 022700 000100  
001570 001402  
001572 005000  
001574 104000

TST11: SCOPE  
MOV #000100,R0 :SET UP R0  
LDFPS R0 :LOAD FLOATING POINT STATUS WITH 000100  
CMP #000100,R0 :CHECK R0  
BEQ TSA11 :BRANCH IF OK  
CLR FPS :FPS NOT YET STORED  
HLT :R0 NOT EQUAL TO 000100

001576 170200  
001600 010000  
001602 022700 000100  
001606 001401  
001610 104000

TSA11: STFPS R0 :STORE FLOATING POINT STATUS IN R0  
MOV R0 FPS :SAVE FPS FOR TYPING  
CMP #000100,R0 :CHECK FLOATING POINT STATUS  
BEQ .+4 :BRANCH IF OK  
HLT :FPS NOT EQUAL TO 000100

\*\*\*\*\*  
:TEST 12: TEST LDFPS (LOAD FLOATING POINT STATUS) AND  
: STFPS (STORE FLOATING POINT STATUS).  
: FPS = 000200, SRC/DST = MO-R5  
\*\*\*\*\*

001612 104400  
001614 012705 000200  
001620 170105  
001622 022705 000200  
001626 001402  
001630 005000  
001632 104000

TST12: SCOPE  
MOV #000200,R5 :SET UP R5  
LDFPS R5 :LOAD FLOATING POINT STATUS WITH 000200  
CMP #000200,R5 :CHECK R5  
BEQ TSA12 :BRANCH IF OK  
CLR FPS :FPS NOT YET STORED  
HLT :R5 NOT EQUAL TO 000200

001634 170205  
001636 010500  
001640 022705 000200  
001644 001401  
001646 104000

TSA12: STFPS R5 :STORE FLOATING POINT STATUS IN R5  
MOV R5 FPS :SAVE FPS FOR TYPING  
CMP #000200,R5 :CHECK FLOATING POINT STATUS  
BEQ .+4 :BRANCH IF OK  
HLT :FPS NOT EQUAL TO 000200

\*\*\*\*\*  
:TEST 13: TEST LDFPS (LOAD FLOATING POINT STATUS) AND  
: STFPS (STORE FLOATING POINT STATUS).  
: FPS = 000400, SRC/DST = MO-R2  
\*\*\*\*\*

001650 104400  
001652 012702 000400  
001656 170102  
001660 022702 000400  
001664 001402  
001666 005000  
001670 104000

TST13: SCOPE  
MOV #000400,R2 :SET UP R2  
LDFPS R2 :LOAD FLOATING POINT STATUS WITH 000400  
CMP #000400,R2 :CHECK R2  
BEQ TSA13 :BRANCH IF OK  
CLR FPS :FPS NOT YET STORED  
HLT :R2 NOT EQUAL TO 000400

001672 170202  
001674 010200  
001676 022702 000400  
001700 001401  
001704 104000

TSA13: STFPS R2 :STORE FLOATING POINT STATUS IN R2  
MOV R2 FPS :SAVE FPS FOR TYPING  
CMP #000400,R2 :CHECK FLOATING POINT STATUS  
BEQ .+4 :BRANCH IF OK  
HLT :FPS NOT EQUAL TO 000400

\*\*\*\*\*  
:TEST 14: TEST LDFPS (LOAD FLOATING POINT STATUS) AND  
: STFPS (STORE FLOATING POINT STATUS).  
: FPS = 001000, SRC/DST = MD-R3  
\*\*\*\*\*

001706 104400  
001710 012703 001000  
001714 170103  
001716 022703 001000  
001722 001402  
001724 005000  
001726 104000

TST14: SCOPE  
MOV #001000,R3 :SET UP R3  
LDFPS R3 :LOAD FLOATING POINT STATUS WITH 001000  
CMP #001000,R3 :CHECK R3  
BEQ TSA14 :BRANCH IF OK  
CLR FPS :FPS NOT YET STORED  
HLT :R3 NOT EQUAL TO 001000

001730 170203  
001732 010300  
001734 022703 001000  
001740 001401  
001742 104000

TSA14: STFPS R3 :STORE FLOATING POINT STATUS IN R3  
MOV R3 FPS :SAVE FPS FOR TYPING  
CMP #001000,R3 :CHECK FLOATING POINT STATUS  
BEQ .+4 :BRANCH IF OK  
HLT :FPS NOT EQUAL TO 001000

\*\*\*\*\*  
:TEST 15: TEST LDFPS (LOAD FLOATING POINT STATUS) AND  
: STFPS (STORE FLOATING POINT STATUS).  
: FPS = 002000, SRC/DST = MD-R0  
\*\*\*\*\*

001744 104400  
001746 012700 002000  
001752 170100  
001754 022700 002000  
001760 001402  
001762 005000  
001764 104000

TST15: SCOPE  
MOV #002000,R0 :SET UP R0  
LDFPS R0 :LOAD FLOATING POINT STATUS WITH 002000  
CMP #002000,R0 :CHECK R0  
BEQ TSA15 :BRANCH IF OK  
CLR FPS :FPS NOT YET STORED  
HLT :R0 NOT EQUAL TO 002000

001766 170200  
001770 010000  
001772 022700 002000  
001776 001401  
002000 104000

TSA15: STFPS R0 :STORE FLOATING POINT STATUS IN R0  
MOV R0 FPS :SAVE FPS FOR TYPING  
CMP #002000,R0 :CHECK FLOATING POINT STATUS  
BEQ .+4 :BRANCH IF OK  
HLT :FPS NOT EQUAL TO 002000

\*\*\*\*\*  
:TEST 16: TEST LDFPS (LOAD FLOATING POINT STATUS) AND  
: STFPS (STORE FLOATING POINT STATUS).  
: FPS = 004000, SRC/DST = MD-R4  
\*\*\*\*\*



```

002002 104400
002004 012704 004000
002010 170104
002012 022704 004000
002015 001402
002020 005000
002022 104000

TST16: SCOPE
MOV #004000,R4 :SET UP R4
LDFPS R4 :LOAD FLOATING POINT STATUS WITH 004000
CMP #004000,R4 :CHECK R4
BEQ TSA16 :BRANCH IF OK
CLR FPS :FPS NOT YET STORED
HLT :R4 NOT EQUAL TO 004000

TSA16: STFPS R4 :STORE FLOATING POINT STATUS IN R4
MOV R4,FPS :SAVE FPS FOR TYPING
CMP #004000,R4 :CHECK FLOATING POINT STATUS
BEQ .+4 :BRANCH IF OK
HLT :FPS NOT EQUAL TO 004000

```

```

*****
:TEST 17: TEST LDFPS (LOAD FLOATING POINT STATUS) AND
: STFPS (STORE FLOATING POINT STATUS).
: FPS = 000000, SRC/DST = MO-R3
*****

```

```

002040 104400
002042 012703 010000
002046 170103
002050 022703 010000
002054 001402
002056 005000
002060 104000

TST17: SCOPE
MOV #010000,R3 :SET UP R3
LDFPS R3 :LOAD FLOATING POINT STATUS WITH 010000
CMP #010000,R3 :CHECK R3
BEQ TSA17 :BRANCH IF OK
CLR FPS :FPS NOT YET STORED
HLT :R3 NOT EQUAL TO 010000

TSA17: STFPS R3 :STORE FLOATING POINT STATUS IN R3
MOV R3,FPS :SAVE FPS FOR TYPING
CMP #000000,R3 :CHECK FLOATING POINT STATUS
BEQ .+4 :BRANCH IF OK
HLT :FPS NOT EQUAL TO 000000

```

```

*****
:TEST 20: TEST LDFPS (LOAD FLOATING POINT STATUS) AND
: STFPS (STORE FLOATING POINT STATUS).
: FPS = 000000, SRC/DST = MO-R2
*****

```

```

002076 104400
002100 012702 020000
002104 170102
002106 022702 020000
002112 001402
002114 005000
002116 104000

TST20: SCOPE
MOV #020000,R2 :SET UP R2
LDFPS R2 :LOAD FLOATING POINT STATUS WITH 020000
CMP #020000,R2 :CHECK R2
BEQ TSA20 :BRANCH IF OK
CLR FPS :FPS NOT YET STORED
HLT :R2 NOT EQUAL TO 020000

TSA20: STFPS R2 :STORE FLOATING POINT STATUS IN R2
MOV R2,FPS :SAVE FPS FOR TYPING
CMP #000000,R2 :CHECK FLOATING POINT STATUS
BEQ .+4 :BRANCH IF OK
HLT :FPS NOT EQUAL TO 000000

```

```

*****
:TEST 21:      TEST LDFPS (LOAD FLOATING POINT STATUS) AND
:              STFPS (STORE FLOATING POINT STATUS).
:              FPS = 040000, SRC/DST = MO-R1
*****

```

```

002134 104400
002136 012701 040000
002142 170101
002144 022701 040000
002150 001402
002152 005000
002154 104000

```

```

TST21: SCOPE
        MOV      #040000,R1      :SET UP R1
        LDFPS   R1              :LOAD FLOATING POINT STATUS WITH 040000
        CMP     #040000,R1      :CHECK R1
        BEQ    TSA21           :BRANCH IF OK
        CLR     FPS             :FPS NOT YET STORED
        HLT

```

```

002156 170201
002160 010100
002162 022701 040000
002166 001401
002170 104000

```

```

TSA21: STFPS   R1              :STORE FLOATING POINT STATUS IN R1
        MOV     R1, FPS         :SAVE FPS FOR TYPING
        CMP    #040000,R1      :CHECK FLOATING POINT STATUS
        BEQ    .+4             :BRANCH IF OK
        HLT

```

```

*****
:TEST 22:      TEST LDFPS (LOAD FLOATING POINT STATUS) AND
:              STFPS (STORE FLOATING POINT STATUS).
:              FPS = 100000, SRC/DST = MO-R5
*****

```

```

002172 104400
002174 012705 100000
002200 170105
002202 022705 100000
002206 001402
002210 005000
002212 104000

```

```

TST22: SCOPE
        MOV      #100000,R5     :SET UP R5
        LDFPS   R5              :LOAD FLOATING POINT STATUS WITH 100000
        CMP     #100000,R5     :CHECK R5
        BEQ    TSA22           :BRANCH IF OK
        CLR     FPS             :FPS NOT YET STORED
        HLT

```

```

002214 170205
002216 010500
002220 022705 100000
002224 001401
002226 104000

```

```

TSA22: STFPS   R5              :STORE FLOATING POINT STATUS IN R5
        MOV     R5, FPS         :SAVE FPS FOR TYPING
        CMP    #100000,R5     :CHECK FLOATING POINT STATUS
        BEQ    .+4             :BRANCH IF OK
        HLT

```

```

*****
:TEST 23:      TEST LDFPS (LOAD FLOATING POINT STATUS) AND
:              STFPS (STORE FLOATING POINT STATUS).
:              FPS = 147756, SRC/DST = MO-R3
*****

```

```

002230 104400
002232 012703 177756
002236 170103
002240 022703 177756
002244 001402
002246 005000
002250 104000

```

```

TST23: SCOPE
        MOV      #177756,R3     :SET UP R3
        LDFPS   R3              :LOAD FLOATING POINT STATUS WITH 177756
        CMP     #177756,R3     :CHECK R3
        BEQ    TSA23           :BRANCH IF OK
        CLR     FPS             :FPS NOT YET STORED
        HLT

```

002252 170203  
002254 010300  
002256 022703 147756  
002262 001401  
002264 104000

TSA23: STFPS R3 :STORE FLOATING POINT STATUS IN R3  
MOV R3, FPS :SAVE FPS FOR TYPING  
CMP #147756, R3 :CHECK FLOATING POINT STATUS  
BEQ .+4 :BRANCH IF OK  
HLT :FPS NOT EQUAL TO 147756

\*\*\*\*\*  
:TEST 24: TEST LDFPS (LOAD FLOATING POINT STATUS) AND  
: STFPS (STORE FLOATING POINT STATUS).  
: FPS = 147755, SRC/DST = MD-R2  
\*\*\*\*\*

002266 104400  
002270 012702 177755  
002274 170102  
002276 022702 177755  
002302 001402  
002304 005000  
002306 104000

TST24: SCOPE  
MOV #177755, R2 :SET UP R2  
LDFPS R2 :LOAD FLOATING POINT STATUS WITH 177755  
CMP #177755, R2 :CHECK R2  
BEQ TSA24 :BRANCH IF OK  
CLR FPS :FPS NOT YET STORED  
HLT :R2 NOT EQUAL TO 177755

002310 170202  
002312 010200  
002314 022702 147755  
002320 001401  
002322 104000

TSA24: STFPS R2 :STORE FLOATING POINT STATUS IN R2  
MOV R2, FPS :SAVE FPS FOR TYPING  
CMP #147755, R2 :CHECK FLOATING POINT STATUS  
BEQ .+4 :BRANCH IF OK  
HLT :FPS NOT EQUAL TO 147755

\*\*\*\*\*  
:TEST 25: TEST LDFPS (LOAD FLOATING POINT STATUS) AND  
: STFPS (STORE FLOATING POINT STATUS).  
: FPS = 147753, SRC/DST = MD-R4  
\*\*\*\*\*

002324 104400  
002326 012704 177753  
002332 170104  
002334 022704 177753  
002340 001402  
002342 005000  
002344 104000

TST25: SCOPE  
MOV #177753, R4 :SET UP R4  
LDFPS R4 :LOAD FLOATING POINT STATUS WITH 177753  
CMP #177753, R4 :CHECK R4  
BEQ TSA25 :BRANCH IF OK  
CLR FPS :FPS NOT YET STORED  
HLT :R4 NOT EQUAL TO 177753

002346 170204  
002350 010400  
002352 022704 147753  
002356 001401  
002360 104000

TSA25: STFPS R4 :STORE FLOATING POINT STATUS IN R4  
MOV R4, FPS :SAVE FPS FOR TYPING  
CMP #147753, R4 :CHECK FLOATING POINT STATUS  
BEQ .+4 :BRANCH IF OK  
HLT :FPS NOT EQUAL TO 147753

\*\*\*\*\*  
:TEST 26: TEST LDFPS (LOAD FLOATING POINT STATUS) AND  
: STFPS (STORE FLOATING POINT STATUS).  
: FPS = 147747, SRC/DST = MD-R1  
\*\*\*\*\*

002362 104400  
002364 012701 177747  
002370 170101  
002372 022701 177747  
002376 001402  
002400 005000  
002402 104000

TST26: SCOPE  
MOV #177747,R1 ;SET UP R1  
LDFPS R1 ;LOAD FLOATING POINT STATUS WITH 177747  
CMP #177747,R1 ;CHECK R1  
BEQ TSA26 ;BRANCH IF OK  
CLR FPS ;FPS NOT YET STORED  
HLT ;R1 NOT EQUAL TO 177747

002404 170201  
002406 010100  
002410 022701 147747  
002414 001401  
002416 104000

TSA26: STFPS R1 ;STORE FLOATING POINT STATUS IN R1  
MOV R1,FPS ;SAVE FPS FOR TYPING  
CMP #147747,R1 ;CHECK FLOATING POINT STATUS  
BEQ .+4 ;BRANCH IF OK  
HLT ;FPS NOT EQUAL TO 147747

\*\*\*\*\*  
:TEST 27: TEST LDFPS (LOAD FLOATING POINT STATUS) AND  
: STFPS (STORE FLOATING POINT STATUS).  
: FPS = 147757, SRC/DST = MD-R5  
\*\*\*\*\*

002420 104400  
002422 012705 177757  
002426 170105  
002430 022705 177757  
002434 001402  
002436 005000  
002440 104000

TST27: SCOPE  
MOV #177757,R5 ;SET UP R5  
LDFPS R5 ;LOAD FLOATING POINT STATUS WITH 177757  
CMP #177757,R5 ;CHECK R5  
BEQ TSA27 ;BRANCH IF OK  
CLR FPS ;FPS NOT YET STORED  
HLT ;R5 NOT EQUAL TO 177757

002442 170205  
002444 010500  
002446 022705 147757  
002450 001401  
002454 104000

TSA27: STFPS R5 ;STORE FLOATING POINT STATUS IN R5  
MOV R5,FPS ;SAVE FPS FOR TYPING  
CMP #147757,R5 ;CHECK FLOATING POINT STATUS  
BEQ .+4 ;BRANCH IF OK  
HLT ;FPS NOT EQUAL TO 147757

\*\*\*\*\*  
:TEST 30: TEST LDFPS (LOAD FLOATING POINT STATUS) AND  
: STFPS (STORE FLOATING POINT STATUS).  
: FPS = 147717, SRC/DST = MI-R0  
\*\*\*\*\*

002456 104400  
002460 000401

SCOPE  
BR TST30 ;BRANCH OVER DATA

002462 177717

DAT30: 177717

002464 012700 002462  
002470 170110  
002472 022700 002462  
002476 001404  
002500 010067 176276  
002504 005000  
002506 104001

TST30: MOV #DAT30, R0 ;MOVE ADDRESS OF DATA INTO R0  
LDFPS (R0) ;LOAD FLOATING POINT STATUS WITH 177717  
CMP #DAT30, R0 ;DID R0 REMAIN UNCHANGED?  
BEQ TSA30 ;BRANCH IF OK  
MOV R0,ANS1 ;SAVE R0 FOR TYPING  
CLR FPS ;FPS NOT YET STORED  
HLT+1 ;R0 NOT EQUAL TO #DAT30

002510 012700 001004

TSA30: MOV #ANS2, R0 ;MOVE ADDRESS OF ANS2 INTO R0

# H02

MAINDEC-11-DCFPA-B  
DCFPAB.P11

TEST OF LDFPS, STFPS, SETF, SETD, SETI, SETL, CFCC  
TEST SECTION

MACY11 27(732) 10-SEP-76 11:04 PAGE 20

```
002514 170210          STFPS (R0)          ;STORE FLOATING POINT STATUS IN ANS2
002516 022700 001004  CMP      #ANS2, R0   ;DID R0 REMAIN UNCHANGED?
002522 001405          BEQ      TSB30       ;BRANCH IF OK
002524 010067 176252  MOV      R0, ANS1    ;SAVE R0 FOR TYPING
002530 016700 176250  MOV      ANS2, FPS   ;MOVE ANS2 INTO FPS
002534 104001          HLT+1             ;R0 NOT EQUAL TO #ANS2

002536 016700 176242  TSB30:  MOV      ANS2, FPS   ;MOVE ANS2 INTO FPS
002542 022700 147717  CMP      #147717, FPS ;CHECK FLOATING POINT STATUS
002546 001401          BEQ      .+4        ;BRANCH IF OK
002550 104000          HLT              ;FPS NOT EQUAL TO 147717
```

```
*****
:TEST 31:      TEST LDFPS (LOAD FLOATING POINT STATUS) AND
:              STFPS (STORE FLOATING POINT STATUS).
:              FPS = 147657, SRC/DST = MD-R0
*****
```

```
002552 104400          SCOPE
002554 012700 177657  TST31:  MOV      #177657, R0   ;SET UP R0
002560 170100          LDFPS   R0             ;LOAD FLOATING POINT STATUS WITH 177657
002562 022700 177657  CMP      #177657, R0   ;CHECK R0
002566 001402          BEQ      TSA31       ;BRANCH IF OK
002570 005000          CLR      FPS        ;FPS NOT YET STORED
002572 104000          HLT              ;R0 NOT EQUAL TO 177657

002574 170200          SCOPE
002576 010000          LDFPS   R0             ;LOAD FLOATING POINT STATUS WITH 177657
002600 022700 147657  TSA31:  MOV      R0, FPS      ;STORE FLOATING POINT STATUS IN R0
002604 001401          BEQ      .+4        ;SAVE FPS FOR TYPING
002606 104000          HLT              ;CHECK FLOATING POINT STATUS
:              ;BRANCH IF OK
:              ;FPS NOT EQUAL TO 147657
```

```
*****
:TEST 32:      TEST LDFPS (LOAD FLOATING POINT STATUS) AND
:              STFPS (STORE FLOATING POINT STATUS).
:              FPS = 147557, SRC/DST = M2-R3
*****
```

```
002610 104400          SCOPE
002612 000401          BR      ,TST32       ;BRANCH OVER DATA

002614 177557          DAT32: 177557

002616 012703 002614  TST32:  MOV      #DAT32, R3   ;MOVE ADDRESS OF DATA INTO R3
002622 170123          LDFPS   (R3)+         ;LOAD FLOATING POINT STATUS WITH 177557
002624 022703 002616  CMP      #DAT32+2, R3  ;DID R3 GET INCREMENTED BY 2?
002630 001404          BEQ      TSA32       ;BRANCH IF OK
002632 010367 176144  MOV      R3, ANS1     ;SAVE R3 FOR TYPING
002636 170200          STFPS   FPS          ;STORE FPS FOR TYPING
002640 104001          HLT+1             ;R3 NOT EQUAL TO #DAT32+2

002642 012703 001004  TSA32:  MOV      #ANS2, R3   ;MOVE ADDRESS OF ANS2 INTO R3
002646 170223          STFPS   (R3)+         ;STORE FLOATING POINT STATUS IN ANS2
002650 022703 001006  CMP      #ANS2+2, R3  ;DID R3 GET INCREMENTED BY 2?
```

```

002654 001405          BEQ     TSB32          ;BRANCH IF OK
002656 010367 176120  MOV     R3,      ANS1      ;SAVE R3 FOR TYPING
002662 016700 176116  MOV     ANS2,    FPS       ;MOVE ANS2 INTO FPS
002666 104001          HLT+1          ;R3 NOT EQUAL TO ANS2+2

002670 016700 176110  TSB32: MOV     ANS2,    FPS       ;MOVE ANS2 INTO FPS
002674 022700 147557  CMP     #147557,FPS      ;CHECK FLOATING POINT STATUS
002700 001401          BEQ     .+4           ;BRANCH IF OK
002702 104000          HLT              ;FPS NOT EQUAL TO 147557
    
```

```

*****
:TEST 33:      TEST LDFPS (LOAD FLOATING POINT STATUS) AND
:              STFPS (STORE FLOATING POINT STATUS).
:              FPS = 147357, SRC/DST = M0-R1
*****
    
```

```

002704 104400          SCOPE
002706 012701 177357  TST33: MOV     #177357,R1    ;SET UP R1
002712 170101          LDFPS   R1             ;LOAD FLOATING POINT STATUS WITH 177357
002714 022701 177357  CMP     #177357,R1    ;CHECK R1
002720 001402          BEQ     TSA33        ;BRANCH IF OK
002722 005000          CLR     FPS         ;FPS NOT YET STORED
002724 104000          HLT              ;R1 NOT EQUAL TO 177357

002726 170201          SCOPE
002730 010100          MOV     R1,      FPS       ;STORE FLOATING POINT STATUS IN R1
002732 022701 147357  TSA33: CMP     #147357,R1    ;SAVE FPS FOR TYPING
002736 001401          BEQ     .+4           ;CHECK FLOATING POINT STATUS
002740 104000          HLT              ;BRANCH IF OK
                                ;FPS NOT EQUAL TO 147357
    
```

```

*****
:TEST 34:      TEST LDFPS (LOAD FLOATING POINT STATUS) AND
:              STFPS (STORE FLOATING POINT STATUS).
:              FPS = 146757, SRC/DST = M3-R4
*****
    
```

```

002742 104400          SCOPE
002744 000403          BR      TST34          ;BRANCH OVER DATA

002746 176757          DAT34: 176757
002750 002746          ADA34: DAT34
002752 001004          ADB34: ANS2

002754 012704 002750  TST34: MOV     #ADA34, R4    ;MOVE ADDRESS OF ADDRESS OF DATA INTO R4
002760 170134          LDFPS   @R4+          ;LOAD FLOATING POINT STATUS WITH 176757
002762 022704 002752  CMP     #ADA34+2,R4    ;DID R4 GET INCREMENTED BY 2?
002766 001404          BEQ     TSA34        ;BRANCH IF OK
002770 010467 176006  MOV     R4,      ANS1      ;SAVE R4 FOR TYPING
002774 170200          STFPS   FPS         ;STORE FPS FOR TYPING
002776 104001          HLT+1          ;R4 NOT EQUAL TO #ADA34

003000 012704 002752  TSA34: MOV     #ADB34, R4    ;MOVE ADDRESS OF ADDRESS OF DATA INTO R4
003004 170234          STFPS   @R4+          ;STORE FLOATING POINT STATUS IN ANS2
003006 022704 002754  CMP     #ADB34+2,R4    ;DID R4 GET INCREMENTED BY 2?
    
```

```

003012 001405          BEQ   TSB34          ;BRANCH IF OK
003014 010467 175762  MOV   R4           ;SAVE R4 FOR TYPING
003020 016700 175760  MOV   ANS2, FPS    ;MOVE ANS2 INTO FPS
003024 104001          HLT+1          ;R4 NOT EQUAL TO #ADB34+2

003026 016700 175752  TSB34: MOV  ANS2, FPS    ;MOVE ANS2 INTO FPS
003032 022700 146757  CMP   #146757, FPS ;CHECK FLOATING POINT STATUS
003036 001401          BEQ   .+4          ;BRANCH IF OK
003040 104000          HLT           ;FPS NOT EQUAL TO 146757

```

```

*****
:TEST 35:      TEST LDFPS (LOAD FLOATING POINT STATUS) AND
:              STFPS (STORE FLOATING POINT STATUS).
:              FPS = 145757, SRC/DST = MO-R2
*****

```

```

003042 104400          SCOPE
003044 012702 175757  TST35: MOV   #175757, R2 ;SET UP R2
003050 170102          LDFPS  R2           ;LOAD FLOATING POINT STATUS WITH 175757
003052 022702 175757  CMP   #175757, R2 ;CHECK R2
003056 001402          BEQ   TSA35        ;BRANCH IF OK
003060 005000          CLR   FPS         ;FPS NOT YET STORED
003062 104000          HLT           ;R2 NOT EQUAL TO 175757

003064 170202          TSA35: STFPS  R2           ;STORE FLOATING POINT STATUS IN R2
003066 019200          MOV   R2, FPS      ;SAVE FPS FOR TYPING
003070 022702 145757  CMP   #145757, R2 ;CHECK FLOATING POINT STATUS
003074 001401          BEQ   .+4          ;BRANCH IF OK
003076 104000          HLT           ;FPS NOT EQUAL TO 145757

```

```

*****
:TEST 36:      TEST LDFPS (LOAD FLOATING POINT STATUS) AND
:              STFPS (STORE FLOATING POINT STATUS).
:              FPS = 143757, SRC/DST = M1-R6
*****

```

```

003100 104400          SCOPE
003102 012746 173757  TST36: MOV   #173757, - (6) ;PUSH DATA ONTO THE STACK
003106 010667 175672  MOV   %6, ANS2    ;SAVE R6
003112 170116          LDFPS  (6)        ;LOAD FLOATING POINT STATUS WITH 173757
003114 020667 175664  CMP   %6, ANS2    ;DID R6 REMAIN UNCHANGED?
003120 001405          BEQ   TSA36        ;BRANCH IF OK
003122 010667 175654  MOV   %6, ANS1    ;SAVE R6 FOR TYPING
003126 016706 175652  MOV   ANS2, %6    ;RESTORE R6
003132 104002          HLT+2          ;R6 (ANS1) NOT EQUAL TO ANS2

003134 170216          TSA36: STFPS  (6)        ;STORE FLOATING POINT STATUS ON STACK
003136 020667 175642  CMP   %6, ANS2    ;DID R6 REMAIN UNCHANGED?
003142 001405          BEQ   TSB36        ;BRANCH IF OK
003144 010667 175632  MOV   %6, ANS1    ;SAVE R6 FOR TYPING
003150 016706 175630  MOV   ANS2, %6    ;RESTORE R6
003154 104002          HLT+2          ;R6 (ANS1) NOT EQUAL TO ANS2

003156 012600          TSB36: MOV   (6)+, FPS ;POP FLOATING POINT STATUS

```

K02

```

003160 022700 143757      CMP      #143757,FPS      ;CHECK FLOATING POINT STATUS
003164 001401      BEQ      .+4            ;BRANCH IF OK
003166 104000      HLT                    ;FPS NOT EQUAL TO 143757

```

```

:*****
:TEST 37:      TEST LDFPS (LOAD FLOATING POINT STATUS) AND
:                STFPS (STORE FLOATING POINT STATUS).
:                FPS = 147757, SRC/DST = M2-R6
:*****

```

```

003170 104400      SCOPE
003172 010667 175606      TST37:  MOV      %R6, ANS2      ;SAVE R6
003175 012746 167757      MOV      #167757,        -(6)      ;PUSH DATA ONTO THE STACK
003202 170126      LDFPS    (6)+            ;LOAD FLOATING POINT STATUS WITH 167757
003204 020667 175574      CMP      %R6, ANS2      ;CHECK R6 FOR AUTO-INCREMENT
003210 001405      BEQ      TSA37          ;BRANCH IF OK
003212 010667 175564      MOV      %R6, ANS1      ;SAVE R6 FOR TYPING
003216 016706 175562      MOV      ANS2, %R6      ;RESTORE R6
003222 104002      HLT+2                ;R6 (ANS1) NOT EQUAL TO ANS2

003224 162767 000002 175552  TSA37:  SUB      #2, ANS2          ;R6 SHOULD BE TWO LESS
003232 170246      STFPS   -(6)            ;STORE FLOATING POINT STATUS ON STACK
003234 020667 175544      CMP      %R6, ANS2      ;CHECK R6 FOR AUTO-DECREMENT
003240 001405      BEQ      TSB37          ;BRANCH IF OK
003242 010667 175534      MOV      %R6, ANS1      ;SAVE R6 FOR TYPING
003246 016706 175532      MOV      ANS2, %R6      ;RESTORE R6
003252 104002      HLT+2                ;R6 (ANS1) NOT EQUAL TO ANS2

003254 012600      SCOPE
003256 022700 147757      TSB37:  MOV      (6)+, FPS      ;POP FLOATING POINT STATUS
003262 001401      CMP      #147757,FPS      ;CHECK FLOATING POINT STATUS
003264 104000      BEQ      .+4            ;BRANCH IF OK
                                HLT                    ;FPS NOT EQUAL TO 147757

```

```

:*****
:TEST 40:      TEST LDFPS (LOAD FLOATING POINT STATUS) AND
:                STFPS (STORE FLOATING POINT STATUS).
:                FPS = 147757, SRC/DST = M3-R6
:*****

```

```

003266 104400      SCOPE
003270 000401      BR      TST40          ;BRANCH OVER DATA

003272 157757      DAT40: 157757

003274 010667 175504      TST40:  MOV      %R6, ANS2      ;SAVE R6
003300 012746 003272      MOV      #DAT40, -(6)      ;PUSH ADDRESS OF DATA ONTO STACK
003304 170136      LDFPS   3(6)+            ;LOAD FLOATING POINT STATUS WITH 157757
003306 020667 175472      CMP      %R6, ANS2      ;CHECK R6 FOR AUTO-INCREMENT BY 2
003312 001405      BEQ      TSA40          ;BRANCH IF OK
003314 010667 175462      MOV      %R6, ANS1      ;SAVE R6 FOR TYPING
003320 016706 175460      MOV      ANS2, %R6      ;RESTORE THE STACK POINTER
003324 104002      HLT+2                ;R6 (ANS1) NOT EQUAL TO ANS2

003326 012746 001002      TSA40:  MOV      #ANS1, -(6)      ;PUSH ADDRESS OF ANSWER BUFF ON STACK

```



```

003332 170236          STFPS  2(6)+      ;STORE FLOATING POINT STATUS IN ANS1
003334 016700 175442  MOV     ANS1,FPS  ;MOVE ANS1 INTO FPS
003340 020667 175440  CMP     %6, ANS2  ;CHECK R6 FOR AUTO-INCREMENT BY 2
003344 001405          BEQ     TSB40     ;BRANCH IF OK
003346 010667 175430  MOV     %6, ANS1  ;SAVE R6 FOR TYPING
003352 016706 175426  MOV     ANS2, %6  ;RESTORE THE STACK POINTER
003356 104002          HLT+2     ;R6 (ANS1) NOT EQUAL TO ANS2

003360 022700 147757  TSB40: CMP     #147757,FPS ;CHECK FLOATING POINT STATUS
003364 001401          BEQ     .+4      ;BRANCH IF OK
003366 104000          HLT     ;FPS NOT EQUAL TO 147757

```

```

:*****
:TEST 41:      TEST LDFPS (LOAD FLOATING POINT STATUS) AND
:              STFPS (STORE FLOATING POINT STATUS).
:              FPS = 107757, SRC/DST = M4-R5
:*****

```

```

003370 104400          SCOPE
003372 000401          BR      TST41      ;BRANCH OVER DATA

003374 137757          DAT41: 137757

003376 012705 003376  TST41: MOV     #DAT41+2,R5 ;MOVE ADDRESS PLUS 2 OF DATA INTO R5
003402 170145          LDFPS  -(R5)      ;LOAD FLOATING POINT STATUS WITH 137757
003404 022705 003374  CMP     #DAT41, R5  ;DID R5 GET DECREMENTED BY 2?
003410 001404          BEQ     TSA41     ;BRANCH IF OK
003412 010567 175364  MOV     R5,ANS1    ;SAVE R5 FOR TYPING
003416 170200          STFPS  FPS       ;STORE FPS FOR TYPING
003420 104001          HLT+1     ;R5 NOT EQUAL TO #DAT41

003422 012705 001006  TSA41: MOV     #ANS2+2,R5 ;MOVE ADDRESS PLUS 2 OF ANS2 INTO R5
003426 170245          STFPS  -(R5)      ;STORE FLOATING POINT STATUS IN ANS2
003430 022705 001004  CMP     #ANS2, R5  ;DID R5 GET DECREMENTED BY 2?
003434 001405          BEQ     TSB41     ;BRANCH IF OK
003436 010567 175340  MOV     R5,ANS1    ;SAVE R5 FOR TYPING
003442 016700 175336  MOV     ANS2, FPS  ;MOVE ANS2 INTO FPS
003446 104001          HLT+1     ;R5 NOT EQUAL TO #ANS2

003450 016700 175330  TSB41: MOV     ANS2, FPS ;MOVE ANS2 INTO FPS
003454 022700 107757  CMP     #107757,FPS ;CHECK FLOATING POINT STATUS
003460 001401          BEQ     .+4      ;BRANCH IF OK
003462 104000          HLT     ;FPS NOT EQUAL TO 107757

```

```

:*****
:TEST 42:      TEST LDFPS (LOAD FLOATING POINT STATUS) AND
:              STFPS (STORE FLOATING POINT STATUS).
:              FPS = 047757, SRC/DST = M5-R4
:*****

```

```

003464 104400          SCOPE
003466 000403          BR      TST42      ;BRANCH OVER DATA

003470 077757          DAT42: 077757

```

```

003472 001004      ADB42:  ANS2
003474 003470      ADA42:  DAT42

003476 012704 003476  TST42:  MOV      #ADA42+2,R4      ;MOVE ADDRESS+2 OF ADDRESS OF DATA INTO R4
003502 170154      LDFPS   2-(R4)          ;LOAD FLOATING POINT STATUS WITH 077757
003504 022704 003474      CMP     #ADA42, R4      ;DID R4 GET DECREMENTED BY 2?
003510 001406      BEQ     TSA42          ;BRANCH IF OK
003512 010467 175264      MOV     R4,      ANS1   ;SAVE R4 FOR TYPING
003516 170200      STFPS  FPS           ;STORE FPS FOR TYPING
003520 104001      HLT+1   ;R4 NOT EQUAL TO #ADA42

003522 012704 003474  TSA42:  MOV     #ADB42+2,R4      ;RESET R4
003526 170254      STFPS  2-(R4)          ;STORE FLOATING POINT STATUS IN ANS2
003530 022704 003472      CMP     #ADB42, R4      ;DID R4 GET DECREMENTED BY 2?
003534 001405      BEQ     TSB42          ;BRANCH IF OK
003536 010467 175240      MOV     R4,      ANS1   ;SAVE R4 FOR TYPING
003542 016700 175236      MOV     ANS2,    FPS     ;MOVE ANS2 INTO FPS
003546 104001      HLT+1   ;R4 NOT EQUAL TO #ADB42

003550 016700 175230  TSB42:  MOV     ANS2,    FPS     ;MOVE ANS2 INTO FPS
003554 022700 047757      CMP     #047757,FPS    ;CHECK FLOATING POINT STATUS
003560 001401      BEQ     .+4           ;BRANCH IF OK
003562 104000      HLT           ;FPS NOT EQUAL TO 047757

```

```

*****
:TEST 43:      TEST LDFPS (LOAD FLOATING POINT STATUS) AND
:              STFPS (STORE FLOATING POINT STATUS).
:              FPS = 105252, SRC/DST = M6-R1
*****

```

```

003564 104400      SCOPE
003566 000401      BR      TST43      ;BRANCH OVER DATA

003570 125252      DAT43: 125252

003572 012701 003550  TST43:  MOV     #DAT43-20,R1  ;MOVE ADDRESS MINUS 20 OF DATA INTO R1
003576 170161 000020  LDFPS  20(R1)          ;LOAD FLOATING POINT STATUS WITH 125252
003602 022701 003550  CMP     #DAT43-20,R1    ;DID R1 REMAIN UNCHANGED?
003606 001404      BEQ     TSA43          ;BRANCH IF OK
003610 010167 175166      MOV     R1,      ANS1   ;SAVE R1 FOR TYPING
003614 170200      STFPS  FPS           ;STORE FPS FOR TYPING
003616 104001      HLT+1   ;R1 NOT EQUAL TO #DAT43-20

003620 012701 001413  TSA43:  MOV     #ANS2+407,R1   ;MOVE ADDRESS PLUS 407 OF ANS2 INTO R1
003624 170261 177371  STFPS  -407(R1)        ;STORE FLOATING POINT STATUS IN ANS2
003630 022701 001413  CMP     #ANS2+407,R1   ;DID R1 REMAIN UNCHANGED?
003634 001405      BEQ     TSB43          ;BRANCH IF OK
003636 010167 175140      MOV     R1,      ANS1   ;SAVE R1 FOR TYPING
003642 016700 175136      MOV     ANS2,    FPS     ;MOVE ANS2 INTO FPS
003646 104001      HLT+1   ;R1 NOT EQUAL TO #ANS2+407

003650 016700 175130  TSB43:  MOV     ANS2,    FPS     ;MOVE ANS2 INTO FPS
003654 022700 105252  CMP     #105252,FPS    ;CHECK FLOATING POINT STATUS
003660 001401      BEQ     .+4           ;BRANCH IF OK
003662 104000      HLT           ;FPS NOT EQUAL TO 105252

```

```

*****
:TEST 44:      TEST LDFPS (LOAD FLOATING POINT STATUS) AND
:              STFPS (STORE FLOATING POINT STATUS).
:              FPS = 042505, SRC/DST = M7-R5
*****

```

```

003664 104400
003666 000403
                                SCOPE
                                BR      TST44      ;BRANCH OVER DATA

003670 052505
003672 003670
003674 001002
DAT44: 052505
ADA44: DAT44
ADB44: ANS1

TST44: MOV      #ADA44+35,R5      ;MOVE ADDR.+35 OF ADDR. OF DATA INTO R5
        LDFPS   @-35(R5)         ;LOAD FLOATING POINT STATUS WITH 052505
        CMP     #ADA44+35,R5     ;DID R5 REMAIN UNCHANGED?
        BEQ     .+10             ;BRANCH IF OK
        MOV     R5,ANS1          ;SAVE R5 FOR TYPING
        HLT+1                    ;R5 NOT EQUAL TO #ADA44+35

003722 012705 003727
003726 170275 000076
003732 016700 175044
003736 022705 003576
003742 001403
003744 010567 175032
003750 104001
        MOV     #ADB44-76,R5     ;MOVE ADDR.-76 OF ADDR. OF ANS1 INTO R5
        STFPS   @76(R5)         ;STORE FLOATING POINT STATUS IN ANS1
        MOV     ANS1,FPS        ;MOVE ANS1 INTO FPS
        CMP     #ADB44-76,R5     ;DID R5 REMAIN UNCHANGED?
        BEQ     .+10             ;BRANCH IF OK
        MOV     R5,ANS1          ;SAVE R5 FOR TYPING
        HLT+1                    ;R5 NOT EQUAL TO #ADB44-76

003752 022700 042505
003756 001401
003760 104000
        CMP     #042505,FPS      ;CHECK FLOATING POINT STATUS
        BEQ     .+4              ;BRANCH IF OK
        HLT                                ;FPS NOT EQUAL TO 042505

```

```

*****
:TEST 45:      TEST LDFPS (LOAD FLOATING POINT STATUS) AND
:              STFPS (STORE FLOATING POINT STATUS).
:              FPS = 146314, SRC/DST = M2-R7
*****

```

```

003762 104400
003764 170127 146314
003770 000403
003772 104000
003774 104000
003776 104000
004000 170227
004002 000000
004004 016700 177772
004010 022700 146314
004014 001401
004016 104000
                                SCOPE
TST45: LDFPS   #146314          ;LOAD FLOATING POINT STATUS WITH 146314
        BR      .+10           ;BRANCH OVER PC CATCHER
        HLT                                ;PC FAILURE
        HLT                                ;PC FAILURE
        HLT                                ;PC FAILURE
        STFPS   (PC)+          ;STORE FLOATING POINT STATUS IN .+2
ANR45: 0
        MOV     ANR45, FPS      ;LOCATION FOR ANSWER
        MOV     #146314,FPS    ;MOV STATUS INTO FPS
        CMP     #146314,FPS    ;CHECK FLOATING POINT STATUS
        BEQ     .+4            ;BRANCH IF OK
        HLT                                ;FPS NOT EQUAL TO 146314

```

```

*****

```

:TEST 46: TEST LDFPS (LOAD FLOATING POINT STATUS) AND  
: STFPS (STORE FLOATING POINT STATUS).  
: FPS = 001443, SRC/DST = M3-R7  
:\*\*\*\*\*

004020 104400  
004022 000401

SCOPE  
BR TST46 :BRANCH OVER DATA

004024 031443

DAT46: 031443

004026 170137 004024  
004032 000403  
004034 104000  
004036 104000  
004040 104000  
004042 170237 001002  
004046 016700 174730  
004052 022700 001443  
004056 001401  
004060 104000

TST46: LDFPS 2#DAT46 :LOAD FLOATING POINT STATUS WITH 031443  
BR .+10 :BRANCH OVER PC CATCHER  
HLT :PC FAILURE  
HLT :PC FAILURE  
HLT :PC FAILURE  
STFPS 2#ANS1 :STORE FLOATING POINT STATUS IN ANS1  
MOV ANS1,FPS :MOVE STATUS INTO FPS  
CMP #001443,FPS :CHECK FLOATING POINT STATUS  
BEQ .+4 :BRANCH IF OK  
HLT :FPS NOT EQUAL TO 001443

:\*\*\*\*\*  
:TEST 47: TEST LDFPS (LOAD FLOATING POINT STATUS) AND  
: STFPS (STORE FLOATING POINT STATUS).  
: FPS = 104611, SRC/DST = M6-R7  
:\*\*\*\*\*

004062 104400  
004064 000401

SCOPE  
BR TST47 :BRANCH OVER DATA

004066 114611

DAT47: 114611

004070 170167 177772  
004074 170267 174702  
004100 016700 174676  
004104 022700 104611  
004110 001401  
004112 104000

TST47: LDFPS DAT47 :LOAD FLOATING POINT STATUS WITH 114611  
STFPS ANS1 :STORE FLOATING POINT STATUS IN ANS1  
MOV ANS1,FPS :MOVE STATUS INTO FPS  
CMP #104611,FPS :CHECK FLOATING POINT STATUS  
BEQ .+4 :BRANCH IF OK  
HLT :FPS NOT EQUAL TO 104611

:\*\*\*\*\*  
:TEST 50: TEST LDFPS (LOAD FLOATING POINT STATUS) AND  
: STFPS (STORE FLOATING POINT STATUS).  
: FPS = 043146, SRC/DST = M7-R7  
:\*\*\*\*\*

004114 104400  
004116 000403

SCOPE  
BR TST50 :BRANCH OVER DATA

004120 063146  
004122 004120  
004124 001002

DAT50: 063146  
ADAS0: DAT50  
ADBS0: ANS1

004126 170177 177770  
004132 170277 177766

TST50: LDFPS 3#ADAS0 :LOAD FLOATING POINT STATUS WITH 063146  
STFPS 3#ADBS0 :STORE FLOATING POINT STATUS IN ANS1

004136 016700  
004140 022700  
004144 001401  
004150 104000

174640  
043146

MOV  
CMP  
BEQ  
HLT

ANS1 FPS  
#043146 FPS  
. +4

: MOVE STATUS INTO FPS  
: CHECK FLOATING POINT STATUS  
: BRANCH IF OK  
: FPS NOT EQUAL TO 043146

\*\*\*\*\*  
: TEST S1: TEST LDFPS (LOAD FLOATING POINT STATUS) AND  
: STFPS (STORE FLOATING POINT STATUS).  
: FPS = 107050, SRC/DST = M0-R2  
\*\*\*\*\*

004152 104400  
004154 012702  
004160 170102  
004162 022702  
004166 001402  
004170 005000  
004172 104000

107050  
107050

TST51: SCOPE  
MOV #107050, R2  
LDFPS R2  
CMP #107050, R2  
BEQ TSA51  
CLR FPS  
HLT

: SET UP R2  
: LOAD FLOATING POINT STATUS WITH 107050  
: CHECK R2  
: BRANCH IF OK  
: FPS NOT YET STORED  
: R2 NOT EQUAL TO 107050

004174 170202  
004176 010200  
004200 022702  
004204 001401  
004206 104000

107050

TSA51: STFPS R2  
MOV R2, FPS  
CMP #107050, R2  
BEQ .+4  
HLT

: STORE FLOATING POINT STATUS IN R2  
: SAVE FPS FOR TYPING  
: CHECK FLOATING POINT STATUS  
: BRANCH IF OK  
: FPS NOT EQUAL TO 107050

\*\*\*\*\*  
: TEST S2: TEST LDFPS (LOAD FLOATING POINT STATUS) AND  
: STFPS (STORE FLOATING POINT STATUS).  
: FPS = 040707, SRC/DST = M3-R0  
\*\*\*\*\*

004210 104400  
004212 000403  
004214 070707  
004216 004214  
004220 001004

004216  
004220

SCOPE  
BR TST52  
DAT52: 070707  
ADAS2: DAT52  
ADBS2: ANS2

: BRANCH OVER DATA

004222 012700  
004226 170130  
004230 022700  
004234 001404  
004236 010067  
004242 170200  
004244 104001

004220  
004222

TST52: MOV #ADAS2, R0  
LDFPS 2(R0)+  
CMP #ADAS2+2, R0  
BEQ TSA52  
MOV R0, ANS1  
STFPS FPS  
HLT+1

: MOVE ADDRESS OF ADDRESS OF DATA INTO R0  
: LOAD FLOATING POINT STATUS WITH 070707  
: DID R0 GET INCREMENTED BY 2?  
: BRANCH IF OK  
: SAVE R0 FOR TYPING  
: STORE FPS FOR TYPING  
: R0 NOT EQUAL TO #ADAS2

004246 012700  
004250 170230  
004254 022700  
004258 001405  
004262 010067  
004266 016700  
004272 104001

174514  
174512

TSA52: MOV #ADBS2, R0  
STFPS 2(R0)+  
CMP #ADBS2+2, R0  
BEQ TSA52  
MOV R0, ANS1  
MOV ANS2, FPS  
HLT+1

: MOVE ADDRESS OF ADDRESS OF DATA INTO R0  
: STORE FLOATING POINT STATUS IN ANS2  
: DID R0 GET INCREMENTED BY 2?  
: BRANCH IF OK  
: SAVE R0 FOR TYPING  
: MOVE ANS2 INTO FPS  
: R0 NOT EQUAL TO #ADBS2+2

004274 016700 174504  
004280 022700 040707  
004290 001401  
004306 104000

TSA52: MOV ANS2 FPS :MOVE ANS2 INTO FPS  
CMP #040707,FPS :CHECK FLOATING POINT STATUS  
BEQ .+4 :BRANCH IF OK  
HLT :FPS NOT EQUAL TO 040707

\*\*\*\*\*  
:TEST 53: TEST LDFPS (LOAD FLOATING POINT STATUS) AND  
: STFPS (STORE FLOATING POINT STATUS).  
: FPS = 143414, SRC/DST = M6-R7  
\*\*\*\*\*

004310 104400  
004312 000401

SCOPE  
BR TST53 :BRANCH OVER DATA

004314 143414

DAT53: 143414

004316 170167 177772  
004322 170267 174454  
004328 016700 174450  
004334 022700 143414  
004336 001401  
004340 104000

TST53: LDFPS DAT53 :LOAD FLOATING POINT STATUS WITH 143414  
STFPS ANS1 :STORE FLOATING POINT STATUS IN ANS1  
MOV ANS1,FPS :MOVE STATUS INTO FPS  
CMP #143414,FPS :CHECK FLOATING POINT STATUS  
BEQ .+4 :BRANCH IF OK  
HLT :FPS NOT EQUAL TO 143414

\*\*\*\*\*  
:TEST 54: TEST LDFPS (LOAD FLOATING POINT STATUS) AND  
: STFPS (STORE FLOATING POINT STATUS).  
: FPS = 004343, SRC/DST = M0-R3  
\*\*\*\*\*

004342 104400  
004344 012703 034343  
004350 170103  
004352 022703 034343  
004356 001402  
004360 005000  
004362 104000

TST54: SCOPE  
MOV #034343,R3 :SET UP R3  
LDFPS R3 :LOAD FLOATING POINT STATUS WITH 034343  
CMP #034343,R3 :CHECK R3  
BEQ TSA54 :BRANCH IF OK  
CLR FPS :FPS NOT YET STORED  
HLT :R3 NOT EQUAL TO 034343

004364 170203  
004366 010300  
004370 022703 004343  
004374 001401  
004376 104000

TSA54: STFPS R3 :STORE FLOATING POINT STATUS IN R3  
MOV R3 FPS :SAVE FPS FOR TYPING  
CMP #004343,R3 :CHECK FLOATING POINT STATUS  
BEQ .+4 :BRANCH IF OK  
HLT :FPS NOT EQUAL TO 004343

\*\*\*\*\*  
:TEST 55: TEST LDFPS (LOAD FLOATING POINT STATUS) AND  
: STFPS (STORE FLOATING POINT STATUS).  
: FPS = 141616, SRC/DST = M4-R5  
\*\*\*\*\*

004400 104400  
004402 000401

SCOPE  
BR TST55 :BRANCH OVER DATA

004404 161616

DAT55: 161616

# E03

MAINDEC-11-DCFPA-B  
DCFPAB.P11

TEST OF LDFPS, STFPS, SETF, SETD, SETI, SETL, CFCC  
TEST SECTION

MACY11 27(732) 10-SEP-76 11:04 PAGE 30

```

004406 012705 004406 TST55: MOV #DAT55+2,R5 :MOVE ADDRESS PLUS 2 OF DATA INTO R5
004410 170145 LDFPS -(R5) :LOAD FLOATING POINT STATUS WITH 161616
004414 022705 004404 CMP #DAT55, R5 :DID R5 GET DECREMENTED BY 2?
004420 001404 BEQ TS55 :BRANCH IF OK
004426 010567 174354 MOV R5,ANS1 :SAVE R5 FOR TYPING
004430 170200 STFPS FPS :STORE FPS FOR TYPING
004436 104001 HLT+1 :R5 NOT EQUAL TO #DAT55

004442 012705 001006 TS55: MOV #ANS2+2,R5 :MOVE ADDRESS PLUS 2 OF ANS2 INTO R5
004446 170245 STFPS -(R5) :STORE FLOATING POINT STATUS IN ANS2
004450 022705 001004 CMP #ANS2, R5 :DID R5 GET DECREMENTED BY 2?
004454 001405 BEQ TS55 :BRANCH IF OK
004458 010567 174330 MOV R5,ANS1 :SAVE R5 FOR TYPING
004462 016700 174326 MOV ANS2, FPS :MOVE ANS2 INTO FPS
004466 104001 HLT+1 :R5 NOT EQUAL TO #ANS2

004470 016700 174320 TS55: MOV ANS2, FPS :MOVE ANS2 INTO FPS
004474 022700 141616 CMP #141616,FPS :CHECK FLOATING POINT STATUS
004478 001401 BEQ .+4 :BRANCH IF OK
004482 104000 HLT :FPS NOT EQUAL TO 141616

```

```

*****
:TEST 56: TEST LDFPS (LOAD FLOATING POINT STATUS) AND
: STFPS (STORE FLOATING POINT STATUS).
: FPS = 006141, SRC/DST = M1-R4
*****

```

```

004474 104400 SCOPE
004476 000401 BR TST56 :BRANCH OVER DATA

004500 016141 DAT56: 016141

004502 012704 004500 TST56: MOV #DAT56, R4 :MOVE ADDRESS OF DATA INTO R4
004506 170114 LDFPS (R4) :LOAD FLOATING POINT STATUS WITH 016141
004510 022704 004500 CMP #DAT56, R4 :DID R4 REMAIN UNCHANGED?
004514 001404 BEQ TS56 :BRANCH IF OK
004518 010467 174260 MOV R4,ANS1 :SAVE R4 FOR TYPING
004522 005000 CLR FPS :FPS NOT YET STORED
004524 104001 HLT+1 :R4 NOT EQUAL TO #DAT56

004526 012704 001004 TS56: MOV #ANS2, R4 :MOVE ADDRESS OF ANS2 INTO R4
004530 170214 STFPS (R4) :STORE FLOATING POINT STATUS IN ANS2
004534 022704 001004 CMP #ANS2, R4 :DID R4 REMAIN UNCHANGED?
004538 001405 BEQ TS56 :BRANCH IF OK
004542 010467 174234 MOV R4,ANS1 :SAVE R4 FOR TYPING
004546 016700 174232 MOV ANS2, FPS :MOVE ANS2 INTO FPS
004550 104001 HLT+1 :R4 NOT EQUAL TO #ANS2

004554 016700 174224 TS56: MOV ANS2, FPS :MOVE ANS2 INTO FPS
004558 022700 006141 CMP #006141,FPS :CHECK FLOATING POINT STATUS
004562 001401 BEQ .+4 :BRANCH IF OK
004566 104000 HLT :FPS NOT EQUAL TO 006141

```

```

:*****
:TEST 57:      TEST LDFPS (LOAD FLOATING POINT STATUS) AND
:              STFPS (STORE FLOATING POINT STATUS).
:              FPS = 140340, SRC/DST = M2-R7
:*****
  
```

```

004670 104400
004672 170127 170340
004674 000403
004676 104000
004678 104000
004680 104000
004682 104000
004684 104000
004686 170227
004688 000000
004690 016700 177772
004692 022700 140340
004694 001401
004696 104000
  
```

```

TST57:  SCOPE
        LDFPS #170340 :LOAD FLOATING POINT STATUS WITH 170340
        BR .+10 :BRANCH OVER PC CATCHER
        HLT :PC FAILURE
        HLT :PC FAILURE
        HLT :PC FAILURE
        STFPS (PC)+ :STORE FLOATING POINT STATUS IN .+2
ANS57:  D :LOCATION FOR ANSWER
        MOV ANR57, FPS :MOV STATUS INTO FPS
        CMP #140340, FPS :CHECK FLOATING POINT STATUS
        BEQ .+4 :BRANCH IF OK
        HLT :FPS NOT EQUAL TO 140340
  
```

```

:*****
:TEST 60:      TEST LDFPS (LOAD FLOATING POINT STATUS) AND
:              STFPS (STORE FLOATING POINT STATUS).
:              FPS = 007417, SRC/DST = M5-R3
:*****
  
```

```

004626 104400
004630 000403
004632 007417
004634 001004
004636 004632
004640 012703 004640
004644 170153
004646 022703 004636
004652 001406
004654 010367 174122
004660 170200
004662 104001
004664 012703 004636
004670 170253
004672 022703 004634
004676 001406
004700 010367 174076
004704 016700 174074
004710 104001
004712 016700 174066
004716 022700 007417
004722 001401
004724 104000
  
```

```

        SCOPE
        BR TST60 :BRANCH OVER DATA
DAT60:  007417
ADB60:  ANS2
ADA60:  DAT60
TST60:  MOV #ADA60+2, R3 :MOVE ADDRESS+2 OF ADDRESS OF DATA INTO R3
        LDFPS @-(R3) :LOAD FLOATING POINT STATUS WITH 007417
        CMP #ADA60, R3 :DID R3 GET DECREMENTED BY 2?
        BEQ TSA60 :BRANCH IF OK
        MOV R3, ANS1 :SAVE R3 FOR TYPING
        STFPS FPS :STORE FPS FOR TYPING
        HLT+1 :R3 NOT EQUAL TO #ADA60
TSA60:  MOV #ADB60+2, R3 :RESET R3
        STFPS @-(R3) :STORE FLOATING POINT STATUS IN ANS2
        CMP #ADB60, R3 :DID R3 GET DECREMENTED BY 2?
        BEQ TSB60 :BRANCH IF OK
        MOV R3, ANS1 :SAVE R3 FOR TYPING
        MOV ANS2, FPS :MOVE ANS2 INTO FPS
        HLT+1 :R3 NOT EQUAL TO #ADB60
TSB60:  MOV ANS2, FPS :MOVE ANS2 INTO FPS
        CMP #007417, FPS :CHECK FLOATING POINT STATUS
        BEQ .+4 :BRANCH IF OK
        HLT :FPS NOT EQUAL TO 007417
  
```

:\*\*\*\*\*



G03

:TEST 61: TEST LDFPS (LOAD FLOATING POINT STATUS) AND  
: STFPS (STORE FLOATING POINT STATUS).  
: FPS = 140741, SRC/DST = M7-R7  
:\*\*\*\*\*

004726 104400  
004730 000403

SCOPE  
BR TST61 ;BRANCH OVER DATA

004732 160741  
004734 004732  
004736 001002

DAT61: 160741  
ADAB1: DAT61  
ADSB1: ANS1

004740 170177 177770  
004744 170277 177766  
004750 016700 174026  
004754 022700 140741  
004760 001401  
004762 104000

TST61: LDFPS @ADAB1 ;LOAD FLOATING POINT STATUS WITH 160741  
STFPS @ADSB1 ;STORE FLOATING POINT STATUS IN ANS1  
MOV ANS1,FPS ;MOVE STATUS INTO FPS  
CMP #140741,FPS ;CHECK FLOATING POINT STATUS  
BEQ .+4 ;BRANCH IF OK  
HLT ;FPS NOT EQUAL TO 140741

:\*\*\*\*\*  
:TEST 62: TEST LDFPS (LOAD FLOATING POINT STATUS) AND  
: STFPS (STORE FLOATING POINT STATUS).  
: FPS = 007016, SRC/DST = M2-R2  
:\*\*\*\*\*

004764 104400  
004766 000401

SCOPE  
BR TST62 ;BRANCH OVER DATA

004770 017016

DAT62: 017016

004772 012702 004770  
004776 170122  
005000 022702 004772  
005004 001404  
005006 010267 173770  
005012 170200  
005014 104001

TST62: MOV #DAT62, R2 ;MOVE ADDRESS OF DATA INTO R2  
LDFPS (R2)+ ;LOAD FLOATING POINT STATUS WITH 017016  
CMP #DAT62+2,R2 ;DID R2 GET INCREMENTED BY 2?  
BEQ TSA62 ;BRANCH IF OK  
MOV R2, ANS1 ;SAVE R2 FOR TYPING  
STFPS FPS ;STORE FPS FOR TYPING  
HLT+1 ;R2 NOT EQUAL TO #DAT62+2

005016 012702 001004  
005022 170222  
005024 022702 001006  
005030 001405  
005032 010267 173744  
005036 016700 173742  
005042 104001

TSA62: MOV #ANS2, R2 ;MOVE ADDRESS OF ANS2 INTO R2  
STFPS (R2)+ ;STORE FLOATING POINT STATUS IN ANS2  
CMP #ANS2+2,R2 ;DID R2 GET INCREMENTED BY 2?  
BEQ TSB62 ;BRANCH IF OK  
MOV R2, ANS1 ;SAVE R2 FOR TYPING  
MOV ANS2, FPS ;MOVE ANS2 INTO FPS  
HLT+1 ;R2 NOT EQUAL TO ANS2+2

005044 016700 173734  
005050 022700 007016  
005054 001401  
005056 104000

TSB62: MOV ANS2, FPS ;MOVE ANS2 INTO FPS  
CMP #007016,FPS ;CHECK FLOATING POINT STATUS  
BEQ .+4 ;BRANCH IF OK  
HLT ;FPS NOT EQUAL TO 007016

:\*\*\*\*\*  
:TEST 63: TEST LDFPS (LOAD FLOATING POINT STATUS) AND  
: STFPS (STORE FLOATING POINT STATUS).  
:\*\*\*\*\*

# H03

MAINDEC-11-DOFPA-B  
DOFPAB.P11

TEST OF LDFPS, STFPS, SETF, SETD, SETI, SETL, CFCC  
TEST SECTION

MACY11 27(732) 10-SEP-76 11:04 PAGE 33

\*\*\*\*\*  
: FPS = 141703, SRC/DST = M7-R1  
\*\*\*\*\*

```
005060 104400          SCOPE
005062 000403          BR      TST63          :BRANCH OVER DATA

005064 141703          DAT63: 141703
005066 005064          ADA63: DAT63
005070 001002          ADB63: ANS1

005072 012701 005123    TST63: MOV      #ADA63+35,R1      :MOVE ADDR.+35 OF ADDR. OF DATA INTO R1
005076 170171 177743    LDFPS   @-35(R1)      :LOAD FLOATING POINT STATUS WITH 141703
005102 022701 005123    CMP     #ADA63+35,R1  :DID R1 REMAIN UNCHANGED?
005106 001403          BEQ     .+10          :BRANCH IF OK
005110 010167 173666    MOV     R1,ANS1      :SAVE R1 FOR TYPING
005114 104001          HLT+1   :R1 NOT EQUAL TO #ADA63+35

005116 012701 004772    MOV     #ADB63-76,R1- :MOVE ADDR.-76 OF ADDR. OF ANS1 INTO R1
005122 170271 000076    STFPS  @76(R1)      :STORE FLOATING POINT STATUS IN ANS1
005126 016700 173650    MOV     ANS1,FPS     :MOVE ANS1 INTO FPS
005132 022701 004772    CMP     #ADB63-76,R1  :DID R1 REMAIN UNCHANGED?
005136 001403          BEQ     .+10          :BRANCH IF OK
005140 010167 173636    MOV     R1,ANS1      :SAVE R1 FOR TYPING
005144 104001          HLT+1   :R1 NOT EQUAL TO #ADB63-76

005146 022700 141703    CMP     #141703,FPS   :CHECK FLOATING POINT STATUS
005152 001401          BEQ     .+4          :BRANCH IF OK
005154 104000          HLT     :FPS NOT EQUAL TO 141703
```

\*\*\*\*\*  
:TEST 64: TEST LDFPS (LOAD FLOATING POINT STATUS) AND  
: STFPS (STORE FLOATING POINT STATUS).  
: FPS = 006054, SRC/DST = M3-R7  
\*\*\*\*\*

```
005156 104400          SCOPE
005160 000401          BR      TST64          :BRANCH OVER DATA

005162 036054          DAT64: 036054

005164 170137 005162    TST64: LDFPS   @#DAT64      :LOAD FLOATING POINT STATUS WITH 036054
005170 000403          BR      .+10          :BRANCH OVER PC CATCHER
005172 104000          HLT     :PC FAILURE
005174 104000          HLT     :PC FAILURE
005176 104000          HLT     :PC FAILURE
005200 170237 001002    STFPS  @#ANS1      :STORE FLOATING POINT STATUS IN ANS1
005204 016700 173572    MOV     ANS1,FPS     :MOVE STATUS INTO FPS
005210 022700 006054    CMP     #006054,FPS   :CHECK FLOATING POINT STATUS
005214 001401          BEQ     .+4          :BRANCH IF OK
005216 104000          HLT     :FPS NOT EQUAL TO 006054
```

\*\*\*\*\*  
:TEST 65: TEST LDFPS (LOAD FLOATING POINT STATUS) AND  
: STFPS (STORE FLOATING POINT STATUS).  
\*\*\*\*\*

: FPS = 103607, SRC/DST = M6-R0  
:\*\*\*\*\*

```

005220 104400          SCOPE
005222 000401          BR      TST65          ;BRANCH OVER DATA

005224 103607          DAT65: 103607

005226 012700 005204    TST65: MOV      #DAT65-20,R0    ;MOVE ADDRESS MINUS 20 OF DATA INTO R0
005232 170160 000020    LDFPS      20(R0)              ;LOAD FLOATING POINT STATUS WITH 103607
005236 022700 005204    CMP        #DAT65-20,R0      ;DID R0 REMAIN UNCHANGED?
005242 001404          BEQ        TSA65              ;BRANCH IF OK
005244 010067 173532    MOV        R0,      ANS1      ;SAVE R0 FOR TYPING
005250 170200          STFPS      FPS                ;STORE FPS FOR TYPING
005252 104001          HLT+1                        ;R0 NOT EQUAL TO #DAT65-20

005254 012700 001413    TSA65: MOV      #ANS2+407,R0   ;MOVE ADDRESS PLUS 407 OF ANS2 INTO R0
005260 170260 177371    STFPS      -407(R0)          ;STORE FLOATING POINT STATUS IN ANS2
005264 022700 001413    CMP        #ANS2+407,R0     ;DID R0 REMAIN UNCHANGED?
005270 001405          BEQ        TSB65              ;BRANCH IF OK
005272 010067 173504    MOV        R0,      ANS1      ;SAVE R0 FOR TYPING
005276 016700 173502    MOV        ANS2,    FPS        ;MOVE ANS2 INTO FPS
005302 104001          HLT+1                        ;R0 NOT EQUAL TO #ANS2+407

005304 016700 173474    TSB65: MOV      ANS2,    FPS    ;MOVE ANS2 INTO FPS
005310 022700 103607    CMP        #103607,FPS      ;CHECK FLOATING POINT STATUS
005314 001401          BEQ        .+4                ;BRANCH IF OK
005316 104000          HLT                          ;FPS NOT EQUAL TO 103607

```

:\*\*\*\*\*  
:TEST 66: TEST LDFPS (LOAD FLOATING POINT STATUS) AND  
: STFPS (STORE FLOATING POINT STATUS).  
: FPS = 044150, SRC/DST = M0-R4  
:\*\*\*\*\*

```

005320 104400          SCOPE
005322 012704 074150    TST66: MOV      #074150,R4    ;SET UP R4
005326 170104          LDFPS      R4                ;LOAD FLOATING POINT STATUS WITH 074150
005330 022704 074150    CMP        #074150,R4       ;CHECK R4
005334 001402          BEQ        TSA66              ;BRANCH IF OK
005336 005000          CLR        FPS                ;FPS NOT YET STORED
005340 104000          HLT                          ;R4 NOT EQUAL TO 074150

005342 170204          TSA66: STFPS      R4                ;STORE FLOATING POINT STATUS IN R4
005344 010400          MOV        R4,      FPS        ;SAVE FPS FOR TYPING
005346 022704 044150    CMP        #044150,R4       ;CHECK FLOATING POINT STATUS
005352 001401          BEQ        .+4                ;BRANCH IF OK
005354 104000          HLT                          ;FPS NOT EQUAL TO 044150

```

:\*\*\*\*\*  
:TEST 67: TEST SETF (SET FLOATING MODE)  
: FPS = 000000  
:\*\*\*\*\*

```

005356 104400
005360 170127 000000 TST67: SCOPE LDFPS #000000 :LOAD FLOATING POINT STATUS WITH 000000
005364 170001 SETF :SET FLOATING MODE
005366 170200 STFPS FPS :STORE FLOATING POINT STATUS IN FPS
005370 022700 000000 CMP #000000,FPS :CHECK FLOATING POINT STATUS
005374 001401 BEQ .+4 :BRANCH IF OK
005376 104000 HLT :FPS NOT EQUAL TO 000000

```

```

*****
:TEST 70: TEST SETF (SET FLOATING MODE)
:FPS = 147557
*****

```

```

005400 104400
005402 170127 177757 TST70: SCOPE LDFPS #177757 :LOAD FLOATING POINT STATUS WITH 177757
005406 170001 SETF :SET FLOATING MODE
005410 170200 STFPS FPS :STORE FLOATING POINT STATUS IN FPS
005412 022700 147557 CMP #147557,FPS :CHECK FLOATING POINT STATUS
005416 001401 BEQ .+4 :BRANCH IF OK
005420 104000 HLT :FPS NOT EQUAL TO 147557

```

```

*****
:TEST 71: TEST SETF (SET FLOATING MODE)
:FPS = 105052
*****

```

```

005422 104400
005424 170127 125252 TST71: SCOPE LDFPS #125252 :LOAD FLOATING POINT STATUS WITH 125252
005430 170001 SETF :SET FLOATING MODE
005432 170200 STFPS FPS :STORE FLOATING POINT STATUS IN FPS
005434 022700 105052 CMP #105052,FPS :CHECK FLOATING POINT STATUS
005440 001401 BEQ .+4 :BRANCH IF OK
005442 104000 HLT :FPS NOT EQUAL TO 105052

```

```

*****
:TEST 72: TEST SETF (SET FLOATING MODE)
:FPS = 042505
*****

```

```

005444 104400
005446 170127 052505 TST72: SCOPE LDFPS #052505 :LOAD FLOATING POINT STATUS WITH 052505
005452 170001 SETF :SET FLOATING MODE
005454 170200 STFPS FPS :STORE FLOATING POINT STATUS IN FPS
005456 022700 042505 CMP #042505,FPS :CHECK FLOATING POINT STATUS
005462 001401 BEQ .+4 :BRANCH IF OK
005464 104000 HLT :FPS NOT EQUAL TO 042505

```

```

*****
:TEST 73: TEST SETD (SET DOUBLE PRECISION MODE)
:FPS = 000200
*****

```

# K03

MAINDEC-11-DCFPA-B  
DCFPAB.P11

TEST OF LDFPS, STFPS, SETF, SETD, SETI, SETL, CFCC  
TEST SECTION

MACY11 27(732) 10-SEP-76 11:04 PAGE 36

```
005466 104400
005470 170127 000000
005474 170011
005476 170200
005500 022700 000200
005504 001401
005506 104000

TST73: SCOPE
LDFPS #000000 ;LOAD FLOATING POINT STATUS WITH 000000
SETD ;SET DOUBLE PRECISION MODE
STFPS FPS ;STORE FLOATING POINT STATUS IN FPS
CMP #000200,FPS ;CHECK FLOATING POINT STATUS
BEQ .+4 ;BRANCH IF OK
HLT ;FPS NOT EQUAL TO 000200
```

```
*****
:TEST 74: TEST SETD (SET DOUBLE PRECISION MODE)
:FPS = 147757
*****
```

```
005510 104400
005512 170127 177757
005516 170011
005520 170200
005522 022700 147757
005526 001401
005530 104000

TST74: SCOPE
LDFPS #177757 ;LOAD FLOATING POINT STATUS WITH 177757
SETD ;SET DOUBLE PRECISION MODE
STFPS FPS ;STORE FLOATING POINT STATUS IN FPS
CMP #147757,FPS ;CHECK FLOATING POINT STATUS
BEQ .+4 ;BRANCH IF OK
HLT ;FPS NOT EQUAL TO 147757
```

```
*****
:TEST 75: TEST SETD (SET DOUBLE PRECISION MODE)
:FPS = 105252
*****
```

```
005532 104400
005534 170127 125252
005540 170011
005542 170200
005544 022700 105252
005550 001401
005552 104000

TST75: SCOPE
LDFPS #125252 ;LOAD FLOATING POINT STATUS WITH 125252
SETD ;SET DOUBLE PRECISION MODE
STFPS FPS ;STORE FLOATING POINT STATUS IN FPS
CMP #105252,FPS ;CHECK FLOATING POINT STATUS
BEQ .+4 ;BRANCH IF OK
HLT ;FPS NOT EQUAL TO 105252
```

```
*****
:TEST 76: TEST SETD (SET DOUBLE PRECISION MODE)
:FPS = 042705
*****
```

```
005554 104400
005556 170127 052505
005562 170011
005564 170200
005566 022700 042705
005572 001401
005574 104000

TST76: SCOPE
LDFPS #052505 ;LOAD FLOATING POINT STATUS WITH 052505
SETD ;SET DOUBLE PRECISION MODE
STFPS FPS ;STORE FLOATING POINT STATUS IN FPS
CMP #042705,FPS ;CHECK FLOATING POINT STATUS
BEQ .+4 ;BRANCH IF OK
HLT ;FPS NOT EQUAL TO 042705
```

```
*****
:TEST 77: TEST SETI (SET INTEGER MODE)
:FPS = 000000
*****
```

# L03

MAINDEC-11-DCFPA-B  
DCFPAB.P11

TEST OF LDFPS, STFPS, SETF, SETD, SETI, SETL, CFCC  
TEST SECTION

MACY11 27(732) 10-SEP-76 11:04 PAGE 37

```
005576 104400
005600 170127 000000
005604 170002
005606 170200
005610 022700 000000
005614 001401
005616 104000

TST77: SCOPE
        LDFPS #000000 ;LOAD FLOATING POINT STATUS WITH 000000
        SETI ;SET INTEGER MODE
        STFPS FPS ;STORE FLOATING POINT STATUS IN FPS
        CMP #000000,FPS ;CHECK FLOATING POINT STATUS
        BEQ .+4 ;BRANCH IF OK
        HLT ;FPS NOT EQUAL TO 000000
```

```
*****
:TEST 100: TEST SETI (SET INTEGER MODE)
:          FPS = 147657
*****
```

```
005620 104400
005622 170127 177757
005626 170002
005630 170200
005632 022700 147657
005636 001401
005640 104000

TST100: SCOPE
        LDFPS #177757 ;LOAD FLOATING POINT STATUS WITH 177757
        SETI ;SET INTEGER MODE
        STFPS FPS ;STORE FLOATING POINT STATUS IN FPS
        CMP #147657,FPS ;CHECK FLOATING POINT STATUS
        BEQ .+4 ;BRANCH IF OK
        HLT ;FPS NOT EQUAL TO 147657
```

```
*****
:TEST 101: TEST SETI (SET INTEGER MODE)
:          FPS = 105252
*****
```

```
005642 104400
005644 170127 125252
005650 170002
005652 170200
005654 022700 105252
005660 001401
005662 104000

TST101: SCOPE
        LDFPS #125252 ;LOAD FLOATING POINT STATUS WITH 125252
        SETI ;SET INTEGER MODE
        STFPS FPS ;STORE FLOATING POINT STATUS IN FPS
        CMP #105252,FPS ;CHECK FLOATING POINT STATUS
        BEQ .+4 ;BRANCH IF OK
        HLT ;FPS NOT EQUAL TO 105252
```

```
*****
:TEST 102: TEST SETI (SET INTEGER MODE)
:          FPS = 042405
*****
```

```
005664 104400
005666 170127 052505
005672 170002
005674 170200
005676 022700 042405
005702 001401
005704 104000

TST102: SCOPE
        LDFPS #052505 ;LOAD FLOATING POINT STATUS WITH 052505
        SETI ;SET INTEGER MODE
        STFPS FPS ;STORE FLOATING POINT STATUS IN FPS
        CMP #042405,FPS ;CHECK FLOATING POINT STATUS
        BEQ .+4 ;BRANCH IF OK
        HLT ;FPS NOT EQUAL TO 042405
```

```
*****
:TEST 103: TEST SETL (SET LONG INTEGER MODE)
:          FPS = 000100
*****
```

005706	104400		TST103:	SCOPE		
005710	170127	000000		LDFPS	#000000	:LOAD FLOATING POINT STATUS WITH 000000
005714	170012			SETL		:SET LONG INTEGER MODE
005716	170200			STFPS	FPS	:STORE FLOATING POINT STATUS IN FPS
005720	022700	000100		CMP	#000100,FPS	:CHECK FLOATING POINT STATUS
005724	001401			BEQ	+.4	:BRANCH IF OK
005726	104000			HLT		:FPS NOT EQUAL TO 000100

\*\*\*\*\*  
 :TEST 104: TEST SETL (SET LONG INTEGER MODE)  
 :FPS = 147757  
 \*\*\*\*\*

005730	104400		TST104:	SCOPE		
005732	170127	177757		LDFPS	#177757	:LOAD FLOATING POINT STATUS WITH 177757
005736	170012			SETL		:SET LONG INTEGER MODE
005740	170200			STFPS	FPS	:STORE FLOATING POINT STATUS IN FPS
005742	022700	147757		CMP	#147757,FPS	:CHECK FLOATING POINT STATUS
005746	001401			BEQ	+.4	:BRANCH IF OK
005750	104000			HLT		:FPS NOT EQUAL TO 147757

\*\*\*\*\*  
 :TEST 105: TEST SETL (SET LONG INTEGER MODE)  
 :FPS = 105352  
 \*\*\*\*\*

005752	104400		TST105:	SCOPE		
005754	170127	125252		LDFPS	#125252	:LOAD FLOATING POINT STATUS WITH 125252
005760	170012			SETL		:SET LONG INTEGER MODE
005762	170200			STFPS	FPS	:STORE FLOATING POINT STATUS IN FPS
005764	022700	105352		CMP	#105352,FPS	:CHECK FLOATING POINT STATUS
005770	001401			BEQ	+.4	:BRANCH IF OK
005772	104000			HLT		:FPS NOT EQUAL TO 105352

\*\*\*\*\*  
 :TEST 106: TEST SETL (SET LONG INTEGER MODE)  
 :FPS = 042505  
 \*\*\*\*\*

005774	104400		TST106:	SCOPE		
005776	170127	052505		LDFPS	#052505	:LOAD FLOATING POINT STATUS WITH 052505
006002	170012			SETL		:SET LONG INTEGER MODE
006004	170200			STFPS	FPS	:STORE FLOATING POINT STATUS IN FPS
006006	022700	042505		CMP	#042505,FPS	:CHECK FLOATING POINT STATUS
006012	001401			BEQ	+.4	:BRANCH IF OK
006014	104000			HLT		:FPS NOT EQUAL TO 042505

\*\*\*\*\*  
 :TEST 107: TEST CFCC (COPY FLOATING CONDITION CODES)  
 :FPS = 000000, CONDITION CODES = 00  
 \*\*\*\*\*

```

006016 104400
006020 170127 000000
006024 170000
006026 013767 177776 172746
006034 042767 177760 172740
006042 170200
006044 022700 000000
006050 001401
006052 104000

006054 022767 000000 172720
006062 001401
006064 104000

```

TST107: SCOPE LDFPS #000000 :LOAD FLOATING POINT STATUS  
CFCC :COPY FLOATING CONDITION CODES  
MOV J#PS,ANS1 :GET CPU STATUS  
BIC #177760,ANS1 :SAVE CONDITION CODES  
STFPS FPS :STORE FLOATING POINT STATUS IN FPS  
CMP #000000,FPS :CHECK FLOATING POINT STATUS  
BEQ .+4 :BRANCH IF OK  
HLT :FPS NOT EQUAL TO 000000

CMP #00, ANS1 :CHECK CONDITION CODES  
BEQ .+4 :BRANCH IF OK  
HLT :CONDITION CODES NOT EQUAL TO 00

```

*****
:TEST 110: TEST CFCC (COPY FLOATING CONDITION CODES)
:FPS = 147757, CONDITION CODES = 17
*****

```

```

006066 104400
006070 170127 177757
006074 170000
006076 013767 177776 172676
006104 042767 177760 172670
006112 170200
006114 022700 147757
006120 001401
006122 104000

006124 022767 000017 172650
006132 001401
006134 104000

```

TST110: SCOPE LDFPS #177757 :LOAD FLOATING POINT STATUS  
CFCC :COPY FLOATING CONDITION CODES  
MOV J#PS,ANS1 :GET CPU STATUS  
BIC #177760,ANS1 :SAVE CONDITION CODES  
STFPS FPS :STORE FLOATING POINT STATUS IN FPS  
CMP #147757,FPS :CHECK FLOATING POINT STATUS  
BEQ .+4 :BRANCH IF OK  
HLT :FPS NOT EQUAL TO 147757

CMP #17, ANS1 :CHECK CONDITION CODES  
BEQ .+4 :BRANCH IF OK  
HLT :CONDITION CODES NOT EQUAL TO 17

```

*****
:TEST 111: TEST CFCC (COPY FLOATING CONDITION CODES)
:FPS = 000001, CONDITION CODES = 01
*****

```

```

006136 104400
006140 170127 000001
006144 170000
006146 013767 177776 172626
006154 042767 177760 172620
006162 170200
006164 022700 000001
006170 001401
006172 104000

006174 022767 000001 172600
006202 001401
006204 104000

```

TST111: SCOPE LDFPS #000001 :LOAD FLOATING POINT STATUS  
CFCC :COPY FLOATING CONDITION CODES  
MOV J#PS,ANS1 :GET CPU STATUS  
BIC #177760,ANS1 :SAVE CONDITION CODES  
STFPS FPS :STORE FLOATING POINT STATUS IN FPS  
CMP #000001,FPS :CHECK FLOATING POINT STATUS  
BEQ .+4 :BRANCH IF OK  
HLT :FPS NOT EQUAL TO 000001

CMP #01, ANS1 :CHECK CONDITION CODES  
BEQ .+4 :BRANCH IF OK  
HLT :CONDITION CODES NOT EQUAL TO 01

```

*****

```



:TEST 112: TEST CFCC (COPY FLOATING CONDITION CODES)  
:FPS = 000002, CONDITION CODES = 02  
:\*\*\*\*\*

006206 104400  
006210 170127 000002  
006214 170000  
006216 013767 177776 172556  
006224 042767 177760 172550  
006232 170200  
006234 022700 000002  
006240 001401  
006242 104000

TST112: SCOPE #000002 :LOAD FLOATING POINT STATUS  
LDFPS :COPY FLOATING CONDITION CODES  
CFCC :GET CPU STATUS  
MOV @#PS,ANS1 :SAVE CONDITION CODES  
BIC #177760,ANS1 :STORE FLOATING POINT STATUS IN FPS  
STFPS FPS :CHECK FLOATING POINT STATUS  
CMP #000002,FPS :BRANCH IF OK  
BEQ .+4 :FPS NOT EQUAL TO 000002  
HLT

006244 022767 000002 172530  
006252 001401  
006254 104000

CMP #02, ANS1 :CHECK CONDITION CODES  
BEQ .+4 :BRANCH IF OK  
HLT :CONDITION CODES NOT EQUAL TO 02

:\*\*\*\*\*  
:TEST 113: TEST CFCC (COPY FLOATING CONDITION CODES)  
:FPS = 000004, CONDITION CODES = 04  
:\*\*\*\*\*

006256 104400  
006260 170127 000004  
006264 170000  
006266 013767 177776 172506  
006274 042767 177760 172500  
006302 170200  
006304 022700 000004  
006310 001401  
006312 104000

TST113: SCOPE #000004 :LOAD FLOATING POINT STATUS  
LDFPS :COPY FLOATING CONDITION CODES  
CFCC :GET CPU STATUS  
MOV @#PS,ANS1 :SAVE CONDITION CODES  
BIC #177760,ANS1 :STORE FLOATING POINT STATUS IN FPS  
STFPS FPS :CHECK FLOATING POINT STATUS  
CMP #000004,FPS :BRANCH IF OK  
BEQ .+4 :FPS NOT EQUAL TO 000004  
HLT

006314 022767 000004 172460  
006322 001401  
006324 104000

CMP #04, ANS1 :CHECK CONDITION CODES  
BEQ .+4 :BRANCH IF OK  
HLT :CONDITION CODES NOT EQUAL TO 04

:\*\*\*\*\*  
:TEST 114: TEST CFCC (COPY FLOATING CONDITION CODES)  
:FPS = 000010, CONDITION CODES = 10  
:\*\*\*\*\*

006326 104400  
006330 170127 000010  
006334 170000  
006336 013767 177776 172436  
006344 042767 177760 172430  
006352 170200  
006354 022700 000010  
006360 001401  
006362 104000

TST114: SCOPE #000010 :LOAD FLOATING POINT STATUS  
LDFPS :COPY FLOATING CONDITION CODES  
CFCC :GET CPU STATUS  
MOV @#PS,ANS1 :SAVE CONDITION CODES  
BIC #177760,ANS1 :STORE FLOATING POINT STATUS IN FPS  
STFPS FPS :CHECK FLOATING POINT STATUS  
CMP #000010,FPS :BRANCH IF OK  
BEQ .+4 :FPS NOT EQUAL TO 000010  
HLT

006364 022767 000010 172410  
006372 001401

CMP #10, ANS1 :CHECK CONDITION CODES  
BEQ .+4 :BRANCH IF OK

006374 104000

HLT

:CONDITION CODES NOT EQUAL TO 10

\*\*\*\*\*  
:TEST 115: TEST CFCC (COPY FLOATING CONDITION CODES)  
:FPS = 000000, CONDITION CODES = 00  
\*\*\*\*\*

006376 104400  
006400 170127 000000  
006404 170000  
006406 013767 177776 172366  
006414 042767 177760 172360  
006422 170200  
006424 022700 000000  
006430 001401  
006432 104000

TST115: SCOPE #000000 :LOAD FLOATING POINT STATUS  
LDFPS :COPY FLOATING CONDITION CODES  
CFCC :GET CPU STATUS  
MOV @#PS,ANS1 :SAVE CONDITION CODES  
BIC #177760,ANS1 :STORE FLOATING POINT STATUS IN FPS  
STFPS FPS :CHECK FLOATING POINT STATUS  
CMP #000000,FPS :BRANCH IF OK  
BEQ .+4 :FPS NOT EQUAL TO 000000  
HLT

006434 022767 000000 172340  
006442 001401  
006444 104000

CMP #00, ANS1 :CHECK CONDITION CODES  
BEQ .+4 :BRANCH IF OK  
HLT :CONDITION CODES NOT EQUAL TO 00

\*\*\*\*\*  
:TEST 116: TEST CFCC (COPY FLOATING CONDITION CODES)  
:FPS = 000040, CONDITION CODES = 00  
\*\*\*\*\*

006446 104400  
006450 170127 000040  
006454 170000  
006456 013767 177776 172316  
006464 042767 177760 172310  
006472 170200  
006474 022700 000040  
006500 001401  
006502 104000

TST116: SCOPE #000040 :LOAD FLOATING POINT STATUS  
LDFPS :COPY FLOATING CONDITION CODES  
CFCC :GET CPU STATUS  
MOV @#PS,ANS1 :SAVE CONDITION CODES  
BIC #177760,ANS1 :STORE FLOATING POINT STATUS IN FPS  
STFPS FPS :CHECK FLOATING POINT STATUS  
CMP #000040,FPS :BRANCH IF OK  
BEQ .+4 :FPS NOT EQUAL TO 000040  
HLT

006504 022767 000000 172270  
006512 001401  
006514 104000

CMP #00, ANS1 :CHECK CONDITION CODES  
BEQ .+4 :BRANCH IF OK  
HLT :CONDITION CODES NOT EQUAL TO 00

\*\*\*\*\*  
:TEST 117: TEST CFCC (COPY FLOATING CONDITION CODES)  
:FPS = 000100, CONDITION CODES = 00  
\*\*\*\*\*

006516 104400  
006520 170127 000100  
006524 170000  
006526 013767 177776 172246  
006534 042767 177760 172240  
006542 170200  
006544 022700 000100  
006550 001401

TST117: SCOPE #000100 :LOAD FLOATING POINT STATUS  
LDFPS :COPY FLOATING CONDITION CODES  
CFCC :GET CPU STATUS  
MOV @#PS,ANS1 :SAVE CONDITION CODES  
BIC #177760,ANS1 :STORE FLOATING POINT STATUS IN FPS  
STFPS FPS :CHECK FLOATING POINT STATUS  
CMP #000100,FPS :BRANCH IF OK  
BEQ .+4

```

006552 104000          HLT          :FPS NOT EQUAL TO 000100
006554 022767 000000 172220      CMP      #00,   ANSI   :CHECK CONDITION CODES
006556 001401          BEQ          :BRANCH IF OK
006564 104000          HLT          :CONDITION CODES NOT EQUAL TO 00

```

```

:*****
:TEST 120:      TEST CFCC (COPY FLOATING CONDITION CODES)
:              FPS = 000200,   CONDITION CODES = 00
:*****

```

```

006566 104400          SCOPE
006570 170127 000200      TST120: LDFPS      #000200      :LOAD FLOATING POINT STATUS
006574 170000          CFCC          :COPY FLOATING CONDITION CODES
006576 013767 177776 172176      MOV      @#PS,ANSI   :GET CPU STATUS
006604 042767 177760 172170      BIC      #177760,ANSI :SAVE CONDITION CODES
006612 170200          STFPS      FPS          :STORE FLOATING POINT STATUS IN FPS
006614 022700 000200      CMP      #000200,FPS :CHECK FLOATING POINT STATUS
006620 001401          BEQ          :BRANCH IF OK
006622 104000          HLT          :FPS NOT EQUAL TO 000200

```

```

006624 022767 000000 172150      CMP      #00,   ANSI   :CHECK CONDITION CODES
006632 001401          BEQ          :BRANCH IF OK
006634 104000          HLT          :CONDITION CODES NOT EQUAL TO 00

```

```

:*****
:TEST 121:      TEST CFCC (COPY FLOATING CONDITION CODES)
:              FPS = 000400,   CONDITION CODES = 00
:*****

```

```

006636 104400          SCOPE
006640 170127 000400      TST121: LDFPS      #000400      :LOAD FLOATING POINT STATUS
006644 170000          CFCC          :COPY FLOATING CONDITION CODES
006646 013767 177776 172126      MOV      @#PS,ANSI   :GET CPU STATUS
006654 042767 177760 172120      BIC      #177760,ANSI :SAVE CONDITION CODES
006662 170200          STFPS      FPS          :STORE FLOATING POINT STATUS IN FPS
006664 022700 000400      CMP      #000400,FPS :CHECK FLOATING POINT STATUS
006670 001401          BEQ          :BRANCH IF OK
006672 104000          HLT          :FPS NOT EQUAL TO 000400

```

```

006674 022767 000000 172100      CMP      #00,   ANSI   :CHECK CONDITION CODES
006702 001401          BEQ          :BRANCH IF OK
006704 104000          HLT          :CONDITION CODES NOT EQUAL TO 00

```

```

:*****
:TEST 122:      TEST CFCC (COPY FLOATING CONDITION CODES)
:              FPS = 001000,   CONDITION CODES = 00
:*****

```

```

006706 104400          SCOPE
006710 170127 001000      TST122: LDFPS      #001000      :LOAD FLOATING POINT STATUS
006714 170000          CFCC          :COPY FLOATING CONDITION CODES
006716 013767 177776 172056      MOV      @#PS,ANSI   :GET CPU STATUS

```

E04

MAINDEC-11-DCFPA-S  
DCFPAB.P11

TEST OF LDFPS, STFPS, SETF, SETD, SETI, SETL, CFCC  
TEST SECTION

MACY11 27(732) 10-SEP-76 11:04 PAGE 43

```

006724 042767 177760 172050      BIC      #177760,ANS1      :SAVE CONDITION CODES
006732 170200                      STFPS    FPS           :STORE FLOATING POINT STATUS IN FPS
006734 022700 001000      CMP      #001000,FPS     :CHECK FLOATING POINT STATUS
006740 001401                      BEQ      .+4           :BRANCH IF OK
006742 104000                      HLT                               :FPS NOT EQUAL TO 001000

006744 022767 000000 172030      CMP      #00.      ANSI   :CHECK CONDITION CODES
006750 001401                      BEQ      .+4           :BRANCH IF OK
006754 104000                      HLT                               :CONDITION CODES NOT EQUAL TO 00

```

```

:*****
:TEST 123:      TEST CFCC (COPY FLOATING CONDITION CODES)
:              FPS = 002000,  CONDITION CODES = 00
:*****

```

```

006756 104400                      SCOPE
006760 170127 002000      TST123: LDFPS    #002000      :LOAD FLOATING POINT STATUS
006764 170000                      CFCC      :COPY FLOATING CONDITION CODES
006766 013767 177776 172006      MOV      @#PS,ANS1     :GET CPU STATUS
006774 042767 177760 172000      BIC      #177760,ANS1  :SAVE CONDITION CODES
007002 170200                      STFPS    FPS           :STORE FLOATING POINT STATUS IN FPS
007004 022700 002000      CMP      #002000,FPS   :CHECK FLOATING POINT STATUS
007010 001401                      BEQ      .+4           :BRANCH IF OK
007012 104000                      HLT                               :FPS NOT EQUAL TO 002000

007014 022767 000000 171760      CMP      #00.      ANSI   :CHECK CONDITION CODES
007022 001401                      BEQ      .+4           :BRANCH IF OK
007024 104000                      HLT                               :CONDITION CODES NOT EQUAL TO 00

```

```

:*****
:TEST 124:      TEST CFCC (COPY FLOATING CONDITION CODES)
:              FPS = 004000,  CONDITION CODES = 00
:*****

```

```

007026 104400                      SCOPE
007030 170127 004000      TST124: LDFPS    #004000      :LOAD FLOATING POINT STATUS
007034 170000                      CFCC      :COPY FLOATING CONDITION CODES
007036 013767 177776 171736      MOV      @#PS,ANS1     :GET CPU STATUS
007044 042767 177760 171730      BIC      #177760,ANS1  :SAVE CONDITION CODES
007052 170200                      STFPS    FPS           :STORE FLOATING POINT STATUS IN FPS
007054 022700 004000      CMP      #004000,FPS   :CHECK FLOATING POINT STATUS
007060 001401                      BEQ      .+4           :BRANCH IF OK
007062 104000                      HLT                               :FPS NOT EQUAL TO 004000

007064 022767 000000 171710      CMP      #00.      ANSI   :CHECK CONDITION CODES
007072 001401                      BEQ      .+4           :BRANCH IF OK
007074 104000                      HLT                               :CONDITION CODES NOT EQUAL TO 00

```

```

:*****
:TEST 125:      TEST CFCC (COPY FLOATING CONDITION CODES)
:              FPS = 000000,  CONDITION CODES = 00
:*****

```

```

007076 104400
007100 170127 010000      TST125: SCOPE
007104 170000             LDFPS      #010000      :LOAD FLOATING POINT STATUS
007106 013767 177776 171666   CFCC              :COPY FLOATING CONDITION CODES
007114 042767 177760 171660   MOV      @#PS,ANS1 :GET CPU STATUS
007122 170200             BIC      #177760,ANS1 :SAVE CONDITION CODES
007124 022700 000000             STFPS     FPS        :STORE FLOATING POINT STATUS IN FPS
007130 001401             CMP      #000000,FPS :CHECK FLOATING POINT STATUS
007132 104000             BEQ      .+4         :BRANCH IF OK
                                HLT              :FPS NOT EQUAL TO 000000

```

```

007134 022767 000000 171640   CMP      #00,      ANS1 :CHECK CONDITION CODES
007142 001401             BEQ      .+4         :BRANCH IF OK
007144 104000             HLT              :CONDITION CODES NOT EQUAL TO 00

```

```

:*****
:TEST 126:      TEST CFCC (COPY FLOATING CONDITION CODES)
:      FPS = 000000,      CONDITION CODES = 00
:*****

```

```

007146 104400
007150 170127 020000      TST126: SCOPE
007154 170000             LDFPS      #020000      :LOAD FLOATING POINT STATUS
007156 013767 177776 171616   CFCC              :COPY FLOATING CONDITION CODES
007164 042767 177760 171610   MOV      @#PS,ANS1 :GET CPU STATUS
007172 170200             BIC      #177760,ANS1 :SAVE CONDITION CODES
007174 022700 000000             STFPS     FPS        :STORE FLOATING POINT STATUS IN FPS
007200 001401             CMP      #000000,FPS :CHECK FLOATING POINT STATUS
007202 104000             BEQ      .+4         :BRANCH IF OK
                                HLT              :FPS NOT EQUAL TO 000000

```

```

007204 022767 000000 171570   CMP      #00,      ANS1 :CHECK CONDITION CODES
007212 001401             BEQ      .+4         :BRANCH IF OK
007214 104000             HLT              :CONDITION CODES NOT EQUAL TO 00

```

```

:*****
:TEST 127:      TEST CFCC (COPY FLOATING CONDITION CODES)
:      FPS = 040000,      CONDITION CODES = 00
:*****

```

```

007216 104400
007220 170127 040000      TST127: SCOPE
007224 170000             LDFPS      #040000      :LOAD FLOATING POINT STATUS
007226 013767 177776 171546   CFCC              :COPY FLOATING CONDITION CODES
007234 042767 177760 171540   MOV      @#PS,ANS1 :GET CPU STATUS
007242 170200             BIC      #177760,ANS1 :SAVE CONDITION CODES
007244 022700 040000             STFPS     FPS        :STORE FLOATING POINT STATUS IN FPS
007250 001401             CMP      #040000,FPS :CHECK FLOATING POINT STATUS
007252 104000             BEQ      .+4         :BRANCH IF OK
                                HLT              :FPS NOT EQUAL TO 040000

```

```

007254 022767 000000 171520   CMP      #00,      ANS1 :CHECK CONDITION CODES
007262 001401             BEQ      .+4         :BRANCH IF OK
007264 104000             HLT              :CONDITION CODES NOT EQUAL TO 00

```

```

:*****

```

:TEST 130: TEST CFCC (COPY FLOATING CONDITION CODES)  
: FPS = 100000, CONDITION CODES = 00  
:\*\*\*\*\*

007266 104400  
007270 170127 100000  
007274 170000  
007276 013767 177776 171476  
007304 042767 177760 171470  
007312 170200  
007314 022700 100000  
007320 001401  
007322 104000  
  
007324 022767 000000 171450  
007332 001401  
007334 104000

TST130: SCOPE  
LDFPS #100000 :LOAD FLOATING POINT STATUS  
CFCC :COPY FLOATING CONDITION CODES  
MOV @#PS,ANS1 :GET CPU STATUS  
BIC #177760,ANS1 :SAVE CONDITION CODES  
STFPS FPS :STORE FLOATING POINT STATUS IN FPS  
CMP #100000,FPS :CHECK FLOATING POINT STATUS  
BEQ .+4 :BRANCH IF OK  
HLT :FPS NOT EQUAL TO 100000  
  
CMP #00, ANS1 :CHECK CONDITION CODES  
BEQ .+4 :BRANCH IF OK  
HLT :CONDITION CODES NOT EQUAL TO 00

:\*\*\*\*\*  
:TEST 131: TEST CFCC (COPY FLOATING CONDITION CODES)  
: FPS = 147756, CONDITION CODES = 16  
:\*\*\*\*\*

007336 104400  
007340 170127 177756  
007344 170000  
007346 013767 177776 171426  
007354 042767 177760 171420  
007362 170200  
007364 022700 147756  
007370 001401  
007372 104000  
  
007374 022767 000016 171400  
007402 001401  
007404 104000

TST131: SCOPE  
LDFPS #177756 :LOAD FLOATING POINT STATUS  
CFCC :COPY FLOATING CONDITION CODES  
MOV @#PS,ANS1 :GET CPU STATUS  
BIC #177760,ANS1 :SAVE CONDITION CODES  
STFPS FPS :STORE FLOATING POINT STATUS IN FPS  
CMP #147756,FPS :CHECK FLOATING POINT STATUS  
BEQ .+4 :BRANCH IF OK  
HLT :FPS NOT EQUAL TO 147756  
  
CMP #16, ANS1 :CHECK CONDITION CODES  
BEQ .+4 :BRANCH IF OK  
HLT :CONDITION CODES NOT EQUAL TO 16

:\*\*\*\*\*  
:TEST 132: TEST CFCC (COPY FLOATING CONDITION CODES)  
: FPS = 147755, CONDITION CODES = 15  
:\*\*\*\*\*

007406 104400  
007410 170127 177755  
007414 170000  
007416 013767 177776 171356  
007424 042767 177760 171350  
007432 170200  
007434 022700 147755  
007440 001401  
007442 104000  
  
007444 022767 000015 171330  
007452 001401

TST132: SCOPE  
LDFPS #177755 :LOAD FLOATING POINT STATUS  
CFCC :COPY FLOATING CONDITION CODES  
MOV @#PS,ANS1 :GET CPU STATUS  
BIC #177760,ANS1 :SAVE CONDITION CODES  
STFPS FPS :STORE FLOATING POINT STATUS IN FPS  
CMP #147755,FPS :CHECK FLOATING POINT STATUS  
BEQ .+4 :BRANCH IF OK  
HLT :FPS NOT EQUAL TO 147755  
  
CMP #15, ANS1 :CHECK CONDITION CODES  
BEQ .+4 :BRANCH IF OK

007454 104000

HLT

:CONDITION CODES NOT EQUAL TO 15

\*\*\*\*\*  
:TEST 133: TEST CFCC (COPY FLOATING CONDITION CODES)  
:FPS = 147753, CONDITION CODES = 13  
\*\*\*\*\*

007456 104400  
007460 170127 177753  
007464 170000  
007466 013767 177776 171306  
007474 042767 177760 171300  
007502 170200  
007504 022700 147753  
007510 001401  
007512 104000

TST133: SCOPE  
LDFPS #177753 :LOAD FLOATING POINT STATUS  
CFCC :COPY FLOATING CONDITION CODES  
MOV @#PS,ANS1 :GET CPU STATUS  
BIC #177760,ANS1 :SAVE CONDITION CODES  
STFPS FPS :STORE FLOATING POINT STATUS IN FPS  
CMP #147753,FPS :CHECK FLOATING POINT STATUS  
BEQ .+4 :BRANCH IF OK  
HLT :FPS NOT EQUAL TO 147753

007514 022767 000013 171260  
007522 001401  
007524 104000

CMP #13, ANS1 :CHECK CONDITION CODES  
BEQ .+4 :BRANCH IF OK  
HLT :CONDITION CODES NOT EQUAL TO 13

\*\*\*\*\*  
:TEST 134: TEST CFCC (COPY FLOATING CONDITION CODES)  
:FPS = 147747, CONDITION CODES = 07  
\*\*\*\*\*

007526 104400  
007530 170127 177747  
007534 170000  
007536 013767 177776 171236  
007544 042767 177760 171230  
007552 170200  
007554 022700 147747  
007560 001401  
007562 104000

TST134: SCOPE  
LDFPS #177747 :LOAD FLOATING POINT STATUS  
CFCC :COPY FLOATING CONDITION CODES  
MOV @#PS,ANS1 :GET CPU STATUS  
BIC #177760,ANS1 :SAVE CONDITION CODES  
STFPS FPS :STORE FLOATING POINT STATUS IN FPS  
CMP #147747,FPS :CHECK FLOATING POINT STATUS  
BEQ .+4 :BRANCH IF OK  
HLT :FPS NOT EQUAL TO 147747

007564 022767 000007 171210  
007572 001401  
007574 104000

CMP #07, ANS1 :CHECK CONDITION CODES  
BEQ .+4 :BRANCH IF OK  
HLT :CONDITION CODES NOT EQUAL TO 07

\*\*\*\*\*  
:TEST 135: TEST CFCC (COPY FLOATING CONDITION CODES)  
:FPS = 147757, CONDITION CODES = 17  
\*\*\*\*\*

007576 104400  
007600 170127 177757  
007604 170000  
007606 013767 177776 171166  
007614 042767 177760 171160  
007622 170200  
007624 022700 147757  
007630 001401

TST135: SCOPE  
LDFPS #177757 :LOAD FLOATING POINT STATUS  
CFCC :COPY FLOATING CONDITION CODES  
MOV @#PS,ANS1 :GET CPU STATUS  
BIC #177760,ANS1 :SAVE CONDITION CODES  
STFPS FPS :STORE FLOATING POINT STATUS IN FPS  
CMP #147757,FPS :CHECK FLOATING POINT STATUS  
BEQ .+4 :BRANCH IF OK

```

007632 104000          HLT          ;FPS NOT EQUAL TO 147757
007634 022767 000017 171140    CMP          #17,   ANSI   ;CHECK CONDITION CODES
007642 001401          BEQ          .+4     ;BRANCH IF OK
007644 104000          HLT          ;CONDITION CODES NOT EQUAL TO 17

```

```

:*****
:TEST 136:          TEST CFCC (COPY FLOATING CONDITION CODES)
:                FPS = 147717,   CONDITION CODES = 17
:*****

```

```

007646 104400          SCOPE
007650 170127 177717    TST136: LDFPS      #177717   ;LOAD FLOATING POINT STATUS
007654 170000          CFCC          ;COPY FLOATING CONDITION CODES
007656 013767 177776 171116    MOV          @#PS,ANSI  ;GET CPU STATUS
007664 042767 177760 171110    BIC          #177760,ANSI ;SAVE CONDITION CODES
007672 170200          STFPS      FPS        ;STORE FLOATING POINT STATUS IN FPS
007674 022700 147717    CMP          #147717,FPS ;CHECK FLOATING POINT STATUS
007700 001401          BEQ          .+4     ;BRANCH IF OK
007702 104000          HLT          ;FPS NOT EQUAL TO 147717

007704 022767 000017 171070    CMP          #17,   ANSI   ;CHECK CONDITION CODES
007712 001401          BEQ          .+4     ;BRANCH IF OK
007714 104000          HLT          ;CONDITION CODES NOT EQUAL TO 17

```

```

:*****
:TEST 137:          TEST CFCC (COPY FLOATING CONDITION CODES)
:                FPS = 147657,   CONDITION CODES = 17
:*****

```

```

007716 104400          SCOPE
007720 170127 177657    TST137: LDFPS      #177657   ;LOAD FLOATING POINT STATUS
007724 170000          CFCC          ;COPY FLOATING CONDITION CODES
007726 013767 177776 171046    MOV          @#PS,ANSI  ;GET CPU STATUS
007734 042767 177760 171040    BIC          #177760,ANSI ;SAVE CONDITION CODES
007742 170200          STFPS      FPS        ;STORE FLOATING POINT STATUS IN FPS
007744 022700 147657    CMP          #147657,FPS ;CHECK FLOATING POINT STATUS
007750 001401          BEQ          .+4     ;BRANCH IF OK
007752 104000          HLT          ;FPS NOT EQUAL TO 147657

007754 022767 000017 171020    CMP          #17,   ANSI   ;CHECK CONDITION CODES
007762 001401          BEQ          .+4     ;BRANCH IF OK
007764 104000          HLT          ;CONDITION CODES NOT EQUAL TO 17

```

```

:*****
:TEST 140:          TEST CFCC (COPY FLOATING CONDITION CODES)
:                FPS = 147557,   CONDITION CODES = 17
:*****

```

```

007766 104400          SCOPE
007770 170127 177557    TST140: LDFPS      #177557   ;LOAD FLOATING POINT STATUS
007774 170000          CFCC          ;COPY FLOATING CONDITION CODES
007776 013767 177776 170776    MOV          @#PS,ANSI  ;GET CPU STATUS

```



```

010004 042767 177760 170770      BIC      #177760,ANS1      ;SAVE CONDITION CODES
010012 170200                      STFPS    FPS              ;STORE FLOATING POINT STATUS IN FPS
010014 022700 147557              CMP      #147557,FPS      ;CHECK FLOATING POINT STATUS
010020 001401                      BEQ      .+4              ;BRANCH IF OK
010022 104000                      HLT                          ;FPS NOT EQUAL TO 147557

010024 022767 000017 170750      CMP      #17,      ANS1      ;CHECK CONDITION CODES
010032 001401                      BEQ      .+4              ;BRANCH IF OK
010034 104000                      HLT                          ;CONDITION CODES NOT EQUAL TO 17

```

```

:*****
:TEST 141:      TEST CFCC (COPY FLOATING CONDITION CODES)
:              FPS = 147357,      CONDITION CODES = 17
:*****

```

```

010036 104400                      SCOPE
010040 170127 177357      TST141: LDFPS    #177357      ;LOAD FLOATING POINT STATUS
010044 170000                      CFCC              ;COPY FLOATING CONDITION CODES
010046 013767 177776 170726      MOV      @#PS,ANS1      ;GET CPU STATUS
010054 042767 177760 170720      BIC      #177760,ANS1      ;SAVE CONDITION CODES
010062 170200                      STFPS    FPS              ;STORE FLOATING POINT STATUS IN FPS
010064 022700 147357              CMP      #147357,FPS      ;CHECK FLOATING POINT STATUS
010070 001401                      BEQ      .+4              ;BRANCH IF OK
010072 104000                      HLT                          ;FPS NOT EQUAL TO 147357

010074 022767 000017 170700      CMP      #17,      ANS1      ;CHECK CONDITION CODES
010102 001401                      BEQ      .+4              ;BRANCH IF OK
010104 104000                      HLT                          ;CONDITION CODES NOT EQUAL TO 17

```

```

:*****
:TEST 142:      TEST CFCC (COPY FLOATING CONDITION CODES)
:              FPS = 146757,      CONDITION CODES = 17
:*****

```

```

010106 104400                      SCOPE
010110 170127 176757      TST142: LDFPS    #176757      ;LOAD FLOATING POINT STATUS
010114 170000                      CFCC              ;COPY FLOATING CONDITION CODES
010116 013767 177776 170656      MOV      @#PS,ANS1      ;GET CPU STATUS
010124 042767 177760 170650      BIC      #177760,ANS1      ;SAVE CONDITION CODES
010132 170200                      STFPS    FPS              ;STORE FLOATING POINT STATUS IN FPS
010134 022700 146757              CMP      #146757,FPS      ;CHECK FLOATING POINT STATUS
010140 001401                      BEQ      .+4              ;BRANCH IF OK
010142 104000                      HLT                          ;FPS NOT EQUAL TO 146757

010144 022767 000017 170630      CMP      #17,      ANS1      ;CHECK CONDITION CODES
010152 001401                      BEQ      .+4              ;BRANCH IF OK
010154 104000                      HLT                          ;CONDITION CODES NOT EQUAL TO 17

```

```

:*****
:TEST 143:      TEST CFCC (COPY FLOATING CONDITION CODES)
:              FPS = 145757,      CONDITION CODES = 17
:*****

```

010156	104400			SCOPE			
010160	170127	175757		TST143: LDFPS	#175757		:LOAD FLOATING POINT STATUS
010164	170000			CFCC			:COPY FLOATING CONDITION CODES
010166	013767	177776	170606	MOV	@#PS,ANS1		:GET CPU STATUS
010174	042767	177760	170600	BIC	#177760,ANS1		:SAVE CONDITION CODES
010202	170200			STFPS	FPS		:STORE FLOATING POINT STATUS IN FPS
010204	022700	145757		CMP	#145757,FPS		:CHECK FLOATING POINT STATUS
010210	001401			BEQ	+.4		:BRANCH IF OK
010212	104000			HLT			:FPS NOT EQUAL TO 145757
010214	022767	000017	170560	CMP	#17, ANS1		:CHECK CONDITION CODES
010222	001401			BEQ	+.4		:BRANCH IF OK
010224	104000			HLT			:CONDITION CODES NOT EQUAL TO 17

```

:*****
:TEST 144:      TEST CFCC (COPY FLOATING CONDITION CODES)
:             FPS = 143757,  CONDITION CODES = 17
:*****

```

010226	104400			SCOPE			
010230	170127	173757		TST144: LDFPS	#173757		:LOAD FLOATING POINT STATUS
010234	170000			CFCC			:COPY FLOATING CONDITION CODES
010236	013767	177776	170536	MOV	@#PS,ANS1		:GET CPU STATUS
010244	042767	177760	170530	BIC	#177760,ANS1		:SAVE CONDITION CODES
010252	170200			STFPS	FPS		:STORE FLOATING POINT STATUS IN FPS
010254	022700	143757		CMP	#143757,FPS		:CHECK FLOATING POINT STATUS
010260	001401			BEQ	+.4		:BRANCH IF OK
010262	104000			HLT			:FPS NOT EQUAL TO 143757
010264	022767	000017	170510	CMP	#17, ANS1		:CHECK CONDITION CODES
010272	001401			BEQ	+.4		:BRANCH IF OK
010274	104000			HLT			:CONDITION CODES NOT EQUAL TO 17

```

:*****
:TEST 145:      TEST CFCC (COPY FLOATING CONDITION CODES)
:             FPS = 147757,  CONDITION CODES = 17
:*****

```

010276	104400			SCOPE			
010300	170127	167757		TST145: LDFPS	#167757		:LOAD FLOATING POINT STATUS
010304	170000			CFCC			:COPY FLOATING CONDITION CODES
010306	013767	177776	170466	MOV	@#PS,ANS1		:GET CPU STATUS
010314	042767	177760	170460	BIC	#177760,ANS1		:SAVE CONDITION CODES
010322	170200			STFPS	FPS		:STORE FLOATING POINT STATUS IN FPS
010324	022700	147757		CMP	#147757,FPS		:CHECK FLOATING POINT STATUS
010330	001401			BEQ	+.4		:BRANCH IF OK
010332	104000			HLT			:FPS NOT EQUAL TO 147757
010334	022767	000017	170440	CMP	#17, ANS1		:CHECK CONDITION CODES
010342	001401			BEQ	+.4		:BRANCH IF OK
010344	104000			HLT			:CONDITION CODES NOT EQUAL TO 17

```

:*****

```

:TEST 146: TEST CFCC (COPY FLOATING CONDITION CODES)  
:FPS = 147757, CONDITION CODES = 17  
:\*\*\*\*\*

010346 104400  
010350 170127 157757  
010354 170000  
010356 013767 177776 170416  
010364 042767 177760 170410  
010372 170200  
010374 022700 147757  
010400 001401  
010402 104000  
  
010404 022767 000017 170370  
010412 001401  
010414 104000

TST146: SCOPE #157757 ;LOAD FLOATING POINT STATUS  
LDFPS ;COPY FLOATING CONDITION CODES  
CFCC ;GET CPU STATUS  
MOV @#PS,ANS1 ;SAVE CONDITION CODES  
BIC #177760,ANS1 ;STORE FLOATING POINT STATUS IN FPS  
STFPS FPS ;CHECK FLOATING POINT STATUS  
CMP #147757,FPS ;BRANCH IF OK  
BEQ .+4 ;FPS NOT EQUAL TO 147757  
HLT  
  
CMP #17, ANS1 ;CHECK CONDITION CODES  
BEQ .+4 ;BRANCH IF OK  
HLT ;CONDITION CODES NOT EQUAL TO 17

:\*\*\*\*\*  
:TEST 147: TEST CFCC (COPY FLOATING CONDITION CODES)  
:FPS = 107757, CONDITION CODES = 17  
:\*\*\*\*\*

010416 104400  
010420 170127 137757  
010424 170000  
010426 013767 177776 170346  
010434 042767 177760 170340  
010442 170200  
010444 022700 107757  
010450 001401  
010452 104000  
  
010454 022767 000017 170320  
010462 001401  
010464 104000

TST147: SCOPE #137757 ;LOAD FLOATING POINT STATUS  
LDFPS ;COPY FLOATING CONDITION CODES  
CFCC ;GET CPU STATUS  
MOV @#PS,ANS1 ;SAVE CONDITION CODES  
BIC #177760,ANS1 ;STORE FLOATING POINT STATUS IN FPS  
STFPS FPS ;CHECK FLOATING POINT STATUS  
CMP #107757,FPS ;BRANCH IF OK  
BEQ .+4 ;FPS NOT EQUAL TO 107757  
HLT  
  
CMP #17, ANS1 ;CHECK CONDITION CODES  
BEQ .+4 ;BRANCH IF OK  
HLT ;CONDITION CODES NOT EQUAL TO 17

:\*\*\*\*\*  
:TEST 150: TEST C. CC (COPY FLOATING CONDITION CODES)  
:FPS = 047757, CONDITION CODES = 17  
:\*\*\*\*\*

010466 104400  
010470 170127 077757  
010474 170000  
010476 013767 177776 170276  
010504 042767 177760 170270  
010512 170200  
010514 022700 047757  
010520 001401  
010522 104000  
  
010524 022767 000017 170250  
010532 001401

TST150: SCOPE #077757 ;LOAD FLOATING POINT STATUS  
LDFPS ;COPY FLOATING CONDITION CODES  
CFCC ;GET CPU STATUS  
MOV @#PS,ANS1 ;SAVE CONDITION CODES  
BIC #177760,ANS1 ;STORE FLOATING POINT STATUS IN FPS  
STFPS FPS ;CHECK FLOATING POINT STATUS  
CMP #047757,FPS ;BRANCH IF OK  
BEQ .+4 ;FPS NOT EQUAL TO 047757  
HLT  
  
CMP #17, ANS1 ;CHECK CONDITION CODES  
BEQ .+4 ;BRANCH IF OK

010534 104000

HLT

;CONDITION CODES NOT EQUAL TO 17

\*\*\*\*\*  
:TEST 151: TEST CFCC (COPY FLOATING CONDITION CODES)  
:FPS = 105252, CONDITION CODES = 12  
\*\*\*\*\*

010536 104400  
010540 170127 125252  
010544 170000  
010546 013767 177776 170226  
010554 042767 177760 170220  
010562 170200  
010564 022700 105252  
010570 001401  
010572 104000

TST151: SCOPE  
LDFPS #125252  
CFCC  
MOV @#PS,ANS1  
BIC #177760,ANS1  
STFPS FPS  
CMP #105252,FPS  
BEQ .+4  
HLT

;LOAD FLOATING POINT STATUS  
;COPY FLOATING CONDITION CODES  
;GET CPU STATUS  
;SAVE CONDITION CODES  
;STORE FLOATING POINT STATUS IN FPS  
;CHECK FLOATING POINT STATUS  
;BRANCH IF OK  
;FPS NOT EQUAL TO 105252

010574 022767 000012 170200  
010602 001401  
010604 104000

CMP #12, ANS1  
BEQ .+4  
HLT

;CHECK CONDITION CODES  
;BRANCH IF OK  
;CONDITION CODES NOT EQUAL TO 12

\*\*\*\*\*  
:TEST 152: TEST CFCC (COPY FLOATING CONDITION CODES)  
:FPS = 042505, CONDITION CODES = 05  
\*\*\*\*\*

010606 104400  
010610 170127 052505  
010614 170000  
010616 013767 177776 170156  
010624 042767 177760 170150  
010632 170200  
010634 022700 042505  
010640 001401  
010642 104000

TST152: SCOPE  
LDFPS #052505  
CFCC  
MOV @#PS,ANS1  
BIC #177760,ANS1  
STFPS FPS  
CMP #042505,FPS  
BEQ .+4  
HLT

;LOAD FLOATING POINT STATUS  
;COPY FLOATING CONDITION CODES  
;GET CPU STATUS  
;SAVE CONDITION CODES  
;STORE FLOATING POINT STATUS IN FPS  
;CHECK FLOATING POINT STATUS  
;BRANCH IF OK  
;FPS NOT EQUAL TO 042505

010644 022767 000005 170130  
010652 001401  
010654 104000

CMP #05, ANS1  
BEQ .+4  
HLT

;CHECK CONDITION CODES  
;BRANCH IF OK  
;CONDITION CODES NOT EQUAL TO 05

\*\*\*\*\*  
:TEST 153: TEST CFCC (COPY FLOATING CONDITION CODES)  
:FPS = 146314, CONDITION CODES = 14  
\*\*\*\*\*

010656 104400  
010660 170127 146314  
010664 170000  
010666 013767 177776 170106  
010674 042767 177760 170100  
010702 170200  
010704 022700 146314  
010710 001401

TST153: SCOPE  
LDFPS #146314  
CFCC  
MOV @#PS,ANS1  
BIC #177760,ANS1  
STFPS FPS  
CMP #146314,FPS  
BEQ .+4

;LOAD FLOATING POINT STATUS  
;COPY FLOATING CONDITION CODES  
;GET CPU STATUS  
;SAVE CONDITION CODES  
;STORE FLOATING POINT STATUS IN FPS  
;CHECK FLOATING POINT STATUS  
;BRANCH IF OK

```

010712 104000          HLT          ;FPS NOT EQUAL TO 146314
010714 022767 000014 170060      CMP      #14,   ANSI  ;CHECK CONDITION CODES
010722 001401          BEQ      .+4     ;BRANCH IF OK
010724 104000          HLT          ;CONDITION CODES NOT EQUAL TO 14

```

```

:*****
:TEST 154:      TEST CFCC (COPY FLOATING CONDITION CODES)
:              FPS = 001443,   CONDITION CODES = 03
:*****

```

```

010726 104400          SCOPE
010730 170127 031443      TST154: LDFPS      #031443      ;LOAD FLOATING POINT STATUS
010734 170000          CFCC          ;COPY FLOATING CONDITION CODES
010736 013767 177776 170036      MOV      @#PS,ANSI  ;GET CPU STATUS
010744 042767 177760 170030      BIC      #177760,ANSI ;SAVE CONDITION CODES
010752 170200          STFPS      FPS      ;STORE FLOATING POINT STATUS IN FPS
010754 022700 001443      CMP      #001443,FPS ;CHECK FLOATING POINT STATUS
010760 001401          BEQ      .+4     ;BRANCH IF OK
010762 104000          HLT          ;FPS NOT EQUAL TO 001443

```

```

010764 022767 000003 170010      CMP      #03,   ANSI  ;CHECK CONDITION CODES
010772 001401          BEQ      .+4     ;BRANCH IF OK
010774 104000          HLT          ;CONDITION CODES NOT EQUAL TO 03

```

```

:*****
:TEST 155:      TEST CFCC (COPY FLOATING CONDITION CODES)
:              FPS = 104611,   CONDITION CODES = 11
:*****

```

```

010776 104400          SCOPE
011000 170127 114611      TST155: LDFPS      #114611      ;LOAD FLOATING POINT STATUS
011004 170000          CFCC          ;COPY FLOATING CONDITION CODES
011006 013767 177776 167766      MOV      @#PS,ANSI  ;GET CPU STATUS
011014 042767 177760 167760      BIC      #177760,ANSI ;SAVE CONDITION CODES
011022 170200          STFPS      FPS      ;STORE FLOATING POINT STATUS IN FPS
011024 022700 104611      CMP      #104611,FPS ;CHECK FLOATING POINT STATUS
011030 001401          BEQ      .+4     ;BRANCH IF OK
011032 104000          HLT          ;FPS NOT EQUAL TO 104611

```

```

011034 022767 000011 167740      CMP      #11,   ANSI  ;CHECK CONDITION CODES
011042 001401          BEQ      .+4     ;BRANCH IF OK
011044 104000          HLT          ;CONDITION CODES NOT EQUAL TO 11

```

```

:*****
:TEST 156:      TEST CFCC (COPY FLOATING CONDITION CODES)
:              FPS = 043146,   CONDITION CODES = 06
:*****

```

```

011046 104400          SCOPE
011050 170127 063146      TST156: LDFPS      #063146      ;LOAD FLOATING POINT STATUS
011054 170000          CFCC          ;COPY FLOATING CONDITION CODES
011056 013767 177776 167716      MOV      @#PS,ANSI  ;GET CPU STATUS

```

```

0111064 042767 177760 167710 BIC #177760,ANSI :SAVE CONDITION CODES
0111072 170200 STFPS FPS :STORE FLOATING POINT STATUS IN FPS
0111074 022700 043146 CMP #043146,FPS :CHECK FLOATING POINT STATUS
0111100 001401 BEQ .+4 :BRANCH IF OK
0111102 104000 HLT :FPS NOT EQUAL TO 043146

0111104 022767 000006 167670 CMP #06, ANSI :CHECK CONDITION CODES
0111112 001401 BEQ .+4 :BRANCH IF OK
0111114 104000 HLT :CONDITION CODES NOT EQUAL TO 06

```

```

:*****
:TEST 157: TEST CFCC (COPY FLOATING CONDITION CODES)
: FPS = 107050, CONDITION CODES = 10
:*****

```

```

0111116 104400 TST157: SCOPE
0111120 170127 107050 LDFPS #107050 :LOAD FLOATING POINT STATUS
0111124 170000 CFCC :COPY FLOATING CONDITION CODES
0111126 013767 177776 167646 MOV @#PS,ANSI :GET CPU STATUS
0111134 042767 177760 167640 BIC #177760,ANSI :SAVE CONDITION CODES
0111142 170200 STFPS FPS :STORE FLOATING POINT STATUS IN FPS
0111144 022700 107050 CMP #107050,FPS :CHECK FLOATING POINT STATUS
0111150 001401 BEQ .+4 :BRANCH IF OK
0111152 104000 HLT :FPS NOT EQUAL TO 107050

0111154 022767 000010 167620 CMP #10, ANSI :CHECK CONDITION CODES
0111162 001401 BEQ .+4 :BRANCH IF OK
0111164 104000 HLT :CONDITION CODES NOT EQUAL TO 10

```

```

:*****
:TEST 160: TEST CFCC (COPY FLOATING CONDITION CODES)
: FPS = 040707, CONDITION CODES = 07
:*****

```

```

0111166 104400 TST160: SCOPE
0111170 170127 070707 LDFPS #070707 :LOAD FLOATING POINT STATUS
0111174 170000 CFCC :COPY FLOATING CONDITION CODES
0111176 013767 177776 167576 MOV @#PS,ANSI :GET CPU STATUS
0111204 042767 177760 167570 BIC #177760,ANSI :SAVE CONDITION CODES
0111212 170200 STFPS FPS :STORE FLOATING POINT STATUS IN FPS
0111214 022700 040707 CMP #040707,FPS :CHECK FLOATING POINT STATUS
0111220 001401 BEQ .+4 :BRANCH IF OK
0111222 104000 HLT :FPS NOT EQUAL TO 040707

0111224 022767 000007 167550 CMP #07, ANSI :CHECK CONDITION CODES
0111232 001401 BEQ .+4 :BRANCH IF OK
0111234 104000 HLT :CONDITION CODES NOT EQUAL TO 07

```

```

:*****
:TEST 161: TEST CFCC (COPY FLOATING CONDITION CODES)
: FPS = 143414, CONDITION CODES = 14
:*****

```

```

011236 104400
011240 170127 143414 TST161: SCOPE
011244 170000 LDFPS #143414 :LOAD FLOATING POINT STATUS
011246 013767 177776 167526 CFCC :COPY FLOATING CONDITION CODES
011254 042767 177760 167520 MOV @#PS,ANSI :GET CPU STATUS
011262 170200 BIC #177760,ANSI :SAVE CONDITION CODES
011264 022700 143414 STFPS FPS :STORE FLOATING POINT STATUS IN FPS
011270 001401 CMP #143414,FPS :CHECK FLOATING POINT STATUS
011272 104000 BEQ .+4 :BRANCH IF OK
HLT :FPS NOT EQUAL TO 143414

011274 022767 000014 167500 CMP #14, ANS1 :CHECK CONDITION CODES
011302 001401 BEQ .+4 :BRANCH IF OK
011304 104000 HLT :CONDITION CODES NOT EQUAL TO 14

```

```

:*****
:TEST 162: TEST CFCC (COPY FLOATING CONDITION CODES)
: FPS = 004343, CONDITION CODES = 03
:*****

```

```

011306 104400
011310 170127 034343 TST162: SCOPE
011314 170000 LDFPS #034343 :LOAD FLOATING POINT STATUS
011316 013767 177776 167456 CFCC :COPY FLOATING CONDITION CODES
011324 042767 177760 167450 MOV @#PS,ANSI :GET CPU STATUS
011332 170200 BIC #177760,ANSI :SAVE CONDITION CODES
011334 022700 004343 STFPS FPS :STORE FLOATING POINT STATUS IN FPS
011340 001401 CMP #004343,FPS :CHECK FLOATING POINT STATUS
011342 104000 BEQ .+4 :BRANCH IF OK
HLT :FPS NOT EQUAL TO 004343

011344 022767 000003 167430 CMP #03, ANS1 :CHECK CONDITION CODES
011352 001401 BEQ .+4 :BRANCH IF OK
011354 104000 HLT :CONDITION CODES NOT EQUAL TO 03

```

```

:*****
:TEST 163: TEST CFCC (COPY FLOATING CONDITION CODES)
: FPS = 141616, CONDITION CODES = 16
:*****

```

```

011356 104400
011360 170127 161616 TST163: SCOPE
011364 170000 LDFPS #161616 :LOAD FLOATING POINT STATUS
011366 013767 177776 167406 CFCC :COPY FLOATING CONDITION CODES
011374 042767 177760 167400 MOV @#PS,ANSI :GET CPU STATUS
011402 170200 BIC #177760,ANSI :SAVE CONDITION CODES
011404 022700 141616 STFPS FPS :STORE FLOATING POINT STATUS IN FPS
011410 001401 CMP #141616,FPS :CHECK FLOATING POINT STATUS
011412 104000 BEQ .+4 :BRANCH IF OK
HLT :FPS NOT EQUAL TO 141616

011414 022767 000016 167360 CMP #16, ANS1 :CHECK CONDITION CODES
011422 001401 BEQ .+4 :BRANCH IF OK
011424 104000 HLT :CONDITION CODES NOT EQUAL TO 16

```

011426	104400			DONE:	SCOPE		
011430	032737	002000	177570		BIT	#SW10,2#SWR	:RING THE BELL?
011436	001005				BNE	1\$	:NO!
011440	012767	000207	001246		MOV	#BELL,TYPE	:TYPE A BELL
011446	000004	012714			TYPE	..TYPE	
011452	005046			1\$:	CLR	-(6)	:CLEAR TRACE TRAP
011454	032737	010000	177570		BIT	#SW12,2#SWR	:RUN WITH TRT?
011462	001010				BNE	2\$	
011464	005167	001226			COM	TRPB	
011470	100005				BPL	2\$	
011472	052716	000020			BIS	#20,(6)	:SET TRACE TRAP
011476	012746	011530			MOV	#3\$, -(6)	:JUMP TO START OF TEST
011502	000002				RTI		
011504	012746	011512		2\$:	MOV	#4\$, -(6)	:JUMP TO START OF TEST
011510	000002				RTI		
011512	013700	000042		4\$:	MOV	2#42,R0	:GET MONITOR ADDRESS
011516	001404				BEG	2\$	:IF NONE
011520	004710				JSR	7,(0)	:GO TO MONITOR
011522	000240				NOP		
011524	000240				NOP		
011526	000240				NOP		
011530	000137	000200		3\$:	JMP	2#200	:JUMP TO START OF TEST
011534	000002			YESRT:	RTI		:RETURN TO PROGRAM FROM TRAP
011536	032737	000400	177570	.EMT:	BIT	#SW08,2#SWR	:KILL LDUB OR LOOP ON SPEC. TEST
011544	001404				BEG	1\$	
011546	123767	177570	167224		CMPB	2#SWR,ICNT	:ON RIGHT TEST? *SW7-0*
011554	001437				BEG	OVER	
011556	113703	177570		1\$:	MOVB	2#SWR,R3	:GET UB BITS
011562	170003				LDUB		
011564	032737	040000	177570		BIT	#SW14,2#SWR	:LOOP ON TEST
011572	001026				BNE	KIT	
011574	032737	004000	177570		BIT	#SW11,2#SWR	:KILL ITERATIONS
011602	001012				BNE	SAVLAD	
011604	105767	167171			TSTB	ICNT+1	
011610	001404				BEG	2\$	:BRANCH IF FIRST
011612	126767	001106	167161		CMPB	TIMES,ICNT+1	:DONE?
011620	001013				BNE	KIT	:BRANCH IF NOT
011622	112767	000001	167151	2\$:	MOVB	#1,ICNT+1	:FIRST ITERATION
011630	105267	167144		SAVLAD:	INCB	ICNT	:COUNT TEST NUMBERS
011634	011667	001060			MOV	(6) LAD	:SAVE LOOP ADDRESS
011640	016737	167134	177570		MOV	ICNT,2#DISPLAY	:DISPLAY TEST NO. AND ITERATION COUNT
011646	000002				RTI		:RETURN
011650	105267	167125		KIT:	INCB	ICNT+1	
011654	016737	167120	177570	OVER:	MOV	ICNT,2#DISPLAY	:SET UP DISPLAY
011662	005767	001032			TST	LAD	:FIRST ONE?
011666	001760				BEG	SAVLAD	
011670	016716	001024			MOV	LAD,(6)	:FUDGE RETURN ADDRESS
011674	000002				RTI		:FIXES PS



011676	032737	002000	177570	.TRP:	BIT	#SW10,2#SWR	:BELL ON ERROR?
011704	001405				BEQ	1\$	:NO - SKIP
011706	012767	000207	001000		MOV	#BELL,TYPE	:TYPE A BELL
011714	000004	012714			TYPE	TYPE	
011720	004767	000406		1\$:	JSR	PC,ERROR	:COUNT THE NUMBER OF ERRORS
011724	010446				MOV	R4,-(6)	
011726	032737	020000	177570		BIT	#SW13,2#SWR	:SKIP TYPEOUT IF SET
011734	001072				BNE	4\$	
011736	000004	012662			TYPE	RETURN	
011742	016646	000002			MOV	2(6),-(6)	:PUT ADDRESS OF INSTRUCTION ON STACK
011746	162716	000002			SUB	#2,(6)	
011752	011605				MOV	(6),TTY	:TYPE (6) IN OCTAL
011754	004767	000212			JSR	%7,PRINTR	:TYPE LEADING ZERO'S
011760	000004	012670			TYPE	SPACE+3	
011764	010005				MOV	R0,TTY	:TYPE R0 IN OCTAL
011766	004767	000200			JSR	%7,PRINTR	:TYPE LEADING ZERO'S
011772	000004	012671			TYPE	SPACE+4	
011776	012703	001002			MOV	#ANS1,R3	:ADDRESS OF DATA
012002	113604				MOVB	2(6)+,R4	:AMOUNT OF DATA IN TABLE
012004	001426				BEQ	3\$	
012006	100016				BPL	2\$	:TYPE STACK?
012010	016667	000006	166764		MOV	6(6),ANS1	
012016	016667	000010	166760		MOV	10(6),ANS2	
012024	016667	000012	166754		MOV	12(6),ANS3	
012032	016667	000014	166750		MOV	14(6),ANS4	
012040	042704	177600			BIC	#177600,R4	:CLEAR SIGN
012044	000004	012671		2\$:	TYPE	SPACE+4	
012050	012305				MOV	(3)+,TTY	:TYPE (3)+ IN OCTAL
012052	004767	000114			JSR	%7,PRINTR	:TYPE LEADING ZERO'S
012056	005304				DEC	R4	
012060	001371				BNE	2\$	
012062	005700			3\$:	TST	FPS	
012064	100016				BPL	4\$	
012066	000004	012665			TYPE	SPACE	
012072	170367	166724			STST	FEC	
012076	016705	166720			MOV	FEC,TTY	:TYPE FEC IN OCTAL
012102	004767	000064			JSR	%7,PRINTR	:TYPE LEADING ZERO'S
012106	000004	012670			TYPE	SPACE+3	
012112	016705	166706			MOV	FEA,TTY	:TYPE FEA IN OCTAL
012116	004767	000050			JSR	%7,PRINTR	:TYPE LEADING ZERO'S
012122	012604			4\$:	MOV	(6)+,R4	
012124	005737	177570			TST	2#SWR	:HALT ON ERROR
012130	100001				BPL	+.4	:SKIP IF CONTINUE
012132	000000				HALT		:HALT ON ERROR!
012134	032737	001000	177570		BIT	#SW09,2#SWR	:CHECK FOR INHIBIT LOOP ON ERROR
012142	001001				BNE	+.4	:SKIP IF LOOP ON ERROR
012144	000002				RTI		
012146	105067	166627			CLRB	ICNT+1	
012152	032737	000400	177570		BIT	#SW08,2#SWR	:CHECK FOR LOAD MICROBREAK
012160	001233				BNE	KIT	:BRANCH IF NOT
012162	113703	177570			MOVB	2#SWR,R3	:PUT MICROBREAK ADDRESS IN R3
012166	170003				LDUB		:LOAD MICROBREAK
012170	000627				BR	KIT	:LOOP ON TEST UNTIL NO ERRORS

# F05

MAINDEC-11-DOFPA-B  
DOFPAE.P11

TEST OF LDFPS, STFPS, SETF, SETD, SETI, SETL, CFCC  
OCTAL DUMP OF A WORD

MACY11 27(732) 10-SEP-76 11:04 PAGE 57

```

U 012172 112767 000001 165600 PRINTR: MOV3 #1,4$ ;SET ZERO FILL SWITCH
012200 000402 BR +6
012202 005067 000122 PRINTS: CLR 4$ ;SUPPRESS LEADING ZERO'S
012206 112767 177772 000115 MOV3 #6,4$+1 ;SET COUNT
012214 010446 MOV R4 -(6) ;SAVE R4
012216 012704 012320 MOV #3$,R4 ;SET POINTER TO FIRST ASCII CHAR.
012222 105014 CLRB (4) ;CLEAR FIRST BYTE
012224 000405 BR 2$ ;ROTATE FIRST BIT
012226 105014 1$: CLRB (4) ;CLEAR BYTE OF CHARACTER
012230 006105 ROL TTY ;ROTATE BIT INTO C
012232 106114 ROLB (4) ;PACK IT
012234 006105 ROL TTY ;ROTATE BIT INTO C
012236 106114 ROLB (4) ;PACK IT
012240 006105 2$: ROL TTY ;ROTATE BIT INTO C
012242 106114 ROLB (4) ;PACK IT
012244 105714 TSTB (4)
012246 001402 BEQ +6
012250 105267 000054 INCB 4$
012254 105767 000050 TSTB 4$ ;CHECK FILL SWITCH
012260 001402 BEQ +6
012262 152724 000060 BITB #0,(4)+ ;MAKE INTO ASCII CHAR
012266 105267 000037 INCB 4$+1
012272 001355 BNE 1$ ;REPEAT
012274 022704 012320 CMP #3$,R4
012300 001002 BNE +6
012302 112724 000060 MOV3 #0,(4)+
012306 105014 CLRB (4)
012310 000004 012320 TYPE 3$ ;TYPE IT
012314 012604 MOV (6)+,R4 ;RESTORE R4
012316 000207 RTS PC

012320 000004 3$: .BLKW 4
012330 000000 4$: 0

012332 005267 000364 ERROR: INC ERRORS ;COUNT ERRORS
012336 132737 000001 000041 BITB #1,2$41 ;AUTO MODE?
012344 001412 BEQ 1$ ;NO!
012346 022767 000010 000346 CMP #10,ERRORS ;TOO MANY?
012354 001006 BNE 1$ ;NOT YET
012356 013700 000042 MOV #42,R0 ;GET ADDRESS
012362 001403 BEQ 1$ ;FORGET IT IF ZERO
012364 005037 000042 CLRB #42 ;ZAP 42
012370 004710 CLR PC ;CALL THE MONITOR
012372 000207 1$: RTS PC ;RETURN
    
```

```

012374 012777 012570 000306 POWDWN: MOV #ILLUP, @UPVEC ;SET FOR FAST UP
012402 012777 000340 000302 MOV #340, @UPVEC+2 ;PRIO:7
012410 170246 STFPS -(6) ;GET THE FPS
012412 170011 SETD ;
012414 174046 STD ACC, -(6) ;SAVE AC'S
012416 174146 STD AC1, -(6)
012420 174246 STD AC2, -(6)
012422 174346 STD AC3, -(6)
012424 172404 LDD AC4, ACC
012426 174046 STD ACC, -(6)
012428 172405 LDD AC5, ACC
012430 174046 STD ACC, -(6)
012432 010046 MOV @R0, -(6) ;SAVE REGISTERS
012434 010146 MOV @R1, -(6)
012436 010246 MOV @R2, -(6)
012438 010346 MOV @R3, -(6)
012440 010446 MOV @R4, -(6)
012442 010546 MOV @R5, -(6)
012450 010667 000220 MOV SP, @SAVE6 ;SAVE SP
012454 012777 012464 000226 MOV #POWUP, @UPVEC ;SET UP VECTOR
012462 000000 HALT

012464 016706 000204 POWUP: MOV @SAVE6, SP ;GET SP
012470 005001 CLR R1 ;WAIT LOOP FOR THE TTY
012472 005201 1$: INC R1
012474 001376 BNE 1$
012476 012605 MOV (6)+, @R5 ;GET THE REGISTERS
012500 012604 MOV (6)+, @R4
012502 012603 MOV (6)+, @R3
012504 012602 MOV (6)+, @R2
012506 012601 MOV (6)+, @R1
012510 012600 MOV (6)+, @R0
012512 170011 SETD ;
012514 172426 LDD (6)+, ACC ;RESTORE THE AC'S
012516 174005 STD ACC, @AC5
012520 172426 LDD (6)+, ACC
012522 174004 STD ACC, @AC4
012524 172726 LDD (6)+, @AC3
012526 172626 LDD (6)+, @AC2
012530 172526 LDD (6)+, @AC1
012532 172426 LDD (6)+, ACC
012534 170126 LDFPS (6)+ ;RESTORE FPS
012536 012777 012374 000140 MOV #POWDWN, @DWNVEC ;SET UP THE POWER DOWN VECTOR
012544 012777 000340 000134 MOV #340, @DWNVEC+2
012552 000004 012556 TYPE ..+2 ;.ASCIZ <15><12>"POWER"
012556 000002 RTI

012570 000000 ILLUP: HALT ;THE POWER UP SEQUENCE WAS STARTED
012572 000776 BR -2 ; BEFORE THE POWER DOWN WAS COMPLETE

```

```

012574 010546 .IOT: MOV TTY, -(6) ;SAVE TTY
012576 017605 000002 MOV @2(6), TTY ;GET ADDRESS TO BE TYPED
012602 105715 1$: TSTB (TTY) ;TERMINATOR?
012604 001406 BEQ 2$ ;
012606 112537 177566 MOVB (TTY)+, @#177566 ;LOAD AND TYPE THE CHARACTER
012612 105737 177564 TSTB @#177564 ;IS THE PRINTER READY
012616 100375 BPL -4 ;
012620 000770 BR 1$ ;GET THE NEXT CHARACTER
012622 017646 000002 2$: MOV @2(6), -(6) ;GET ADDRESS TO BE TYPED
012626 062766 000002 000004 ADD #2, 4(6) ;ADD 2 TO THE ADDRESS
012634 022666 000002 CMP (6)+, 2(6) ;IS IT .+2?
012640 001006 3$: SNE 3$ ;NO
012642 062705 000002 ADD #2, TTY ;ADD 2 TO THE ADDRESS
012646 042705 000001 BIC #1, TTY ;BACK UP TO AN EVEN BYTE
012652 010566 000002 MOV TTY, 2(6) ;RESTORE ADDRESS
012656 012605 3$: MOV (6)+, TTY ;RESTORE TTY
012660 000002 RTI ;RETURN

012662 005015 000 RETURN: .ASCIZ <15><12> ;RETURN AND LINEFEED
012665 015 020012 020040 SPACE: .ASCIZ <15><12>" " ;RETURN AND 3 SPACES
012672 000

012674 012674 .EVEN
012676 000000 SAVE6: 0
012700 172160 FPADDR: 172160 ;FLOATING POINT ADDRESS ON THE 11/20
012704 000244 000246 FPVECT: 244, 246 ;FLOATING POINT VECTOR ADDRESS
012710 000024 000026 DWNVEC: 24, 26 ;POWER DOWN VECTOR ADDRESS
012714 000024 000026 UPVEC: 24, 26 ;POWER UP VECTOR ADDRESS
012716 000000 .TYPE: 0
012720 000000 TRPB: 0
012722 000000 LAD: 0 ;LOOP ADDRESS
012724 000377 ERRORS: 0 ;ERROR COUNT
000001 TIMES: 377 ;ITERATION COUNT
.END

```





# K05

MAINDEC-11-DOFPA-B  
DOFPAB.F11

TEST OF LDFPS, STFPS, SETF, SETD, SETI, SETL, CFCC  
CROSS REFERENCE TABLE -- USER SYMBOLS

MACY11 27(732) 10-SEP-76 11:04 PAGE 63

		1556	1564	1569	1581	1582	1583	1589	1611	1619	1624	1645	1665	1673
		1678	1699	1707	1711	1727	1728	1729	1734	1754	1762	1767	1782	1788
		1802	1816	1830	1844	1858	1872	1886	1900	1914	1928	1942	1956	1970
		1984	1998	2012	2028	2032	2048	2052	2068	2072	2088	2092	2108	2112
		2128	2132	2148	2152	2168	2172	2188	2192	2208	2212	2228	2232	2248
		2252	2268	2272	2288	2292	2308	2312	2328	2332	2348	2352	2368	2372
		2388	2392	2408	2412	2428	2432	2448	2452	2468	2472	2488	2492	2508
		2512	2528	2532	2548	2552	2568	2572	2588	2592	2608	2612	2628	2632
		2648	2652	2668	2672	2688	2692	2708	2712	2728	2732	2748	2752	2768
		2772	2788	2792	2808	2812	2828	2832	2848	2852	2868	2872	2888	2892
		2908	2912											
ICNT	001000	416#	450*	2941	2949	2951	2953*	2954*	2956	2959*	2960	3013*		
ILLUP	012570	3065	3112#											
KIT	011650	2946	2952	2959#	3015	3018								
LAD	012720	451*	2955*	2961	2963	3144#								
LDSC	= 170004	401#												
LDUB	= 170003	397#												
MRS	= 170006	400#												
M1120	001054	429	434#											
N	= 000164	365#	452	473#	494#	515#	536#	557#	578#	599#	620#	641#	662#	683#
		704#	725#	746#	767#	788#	809#	830#	851#	872#	893#	914#	935#	968#
		989#	1022#	1043#	1078#	1099#	1128#	1158#	1192#	1225#	1260#	1293#	1326#	1346#
		1369#	1388#	1409#	1430#	1465#	1484#	1505#	1538#	1571#	1591#	1626#	1647#	1680#
		1713#	1736#	1769#	1790#	1804#	1818#	1832#	1846#	1860#	1874#	1888#	1902#	1916#
		1930#	1944#	1958#	1972#	1986#	2000#	2014#	2034#	2054#	2074#	2094#	2114#	2134#
		2154#	2174#	2194#	2214#	2234#	2254#	2274#	2294#	2314#	2334#	2354#	2374#	2394#
		2414#	2434#	2454#	2474#	2494#	2514#	2534#	2554#	2574#	2594#	2614#	2634#	2654#
		2674#	2694#	2714#	2734#	2754#	2774#	2794#	2814#	2834#	2854#	2874#	2894#	2914#
		2942	2960#											
OVER	011654	382#	1339*	1584*	2970*	3049*	3062*	3063*						
PC	=%000007	440	3065#	3107										
POWDWN	012374	3084	3087#											
POWUP	012464	2978	2981	2994	3002	3005	3020#							
PRINTR	012172	3022#												
PRINTS	012202	366#	2023	2043	2063	2083	2103	2123	2143	2163	2183	2203	2223	2243
PS	= 177776	2263	2283	2303	2323	2343	2363	2383	2403	2423	2443	2463	2483	2503
		2523	2543	2563	2583	2603	2623	2643	2663	2683	2703	2723	2743	2763
		2783	2803	2823	2843	2863	2883	2903						
		2974	3133#											
RETURN	012662	374#	443*	460*	461	462	467*	468	469	628*	629	630	635*	636
RO	=%000000	637	712*	713	714	719*	720	721	947*	948	949	951	955*	956*
		957	959	976*	977	978	983*	984	985	1444*	1445	1446	1448	1452*
		1453*	1454	1456	1748*	1749	1750	1752	1756*	1757*	1758	1760	2930*	2980
		3059*	3077	3096*										
R1	=%000001	375#	481*	482	483	488*	489	490	586*	587	588	593*	594	595
		796*	797	798	803*	804	805	901*	902	903	908*	909	910	1030*
		1031	1032	1037*	1038	1039	1272*	1273	1274	1276	1280*	1281*	1282	1284
		1694*	1695	1696	1698	1701*	1702*	1704	1706	3078	3088*	3089*	3095*	
R2	=%000002	376#	502*	503	504	509*	510	511	670*	671	672	677*	678	679
		775*	776	777	782*	783	784	859*	860	861	866*	867	868	1086*
		1087	1088	1093*	1094	1095	1417*	1418	1419	1424*	1425	1426	1659*	1660
		1661	1663	1667*	1668*	1669	1671	3079	3094*					
R3	=%000003	377#	523*	524	525	530*	531	532	691*	692	693	698*	699	700
		754*	755	756	761*	762	763	838*	839	840	845*	846	847	1001*
		1002	1003	1005	1009*	1010*	1011	1013	1492*	1493	1494	1499*	1500	1501
		1605*	1606	1607	1609	1613*	1614*	1615	1617	2943*	2983*	3016*	3080	3093*





TSA30	002510	950	955#
TSA31	002574	979	983#
TSA32	002642	1004	1009#
TSA33	002726	1033	1037#
TSA34	003000	1060	1065#
TSA35	003064	1089	1093#
TSA36	003134	1111	1116#
TSA37	003224	1140	1145#
TSA4	001350	526	530#
TSA40	003326	1174	1179#
TSA41	003422	1207	1212#
TSA42	003526	1242	1248#
TSA43	003620	1275	1280#
TSA5	001406	547	551#
TSA51	004174	1420	1424#
TSA52	004246	1447	1452#
TSA54	004364	1495	1499#
TSA55	004432	1520	1525#
TSA56	004526	1553	1558#
TSA6	001444	568	572#
TSA60	004670	1608	1614#
TSA62	005016	1662	1667#
TSA65	005254	1751	1756#
TSA66	005342	1780	1784#
TSA7	001502	589	593#
TSB30	002536	958	963#
TSB32	002670	1012	1017#
TSB34	003026	1068	1073#
TSB36	003156	1118	1123#
TSB37	003254	1148	1153#
TSB40	003360	1183	1188#
TSB41	003450	1215	1220#
TSB42	003550	1250	1255#
TSB43	003650	1283	1288#
TSB52	004274	1455	1460#
TSB55	004460	1528	1533#
TSB56	004554	1561	1566#
TSB60	004712	1616	1621#
TSB62	005044	1670	1675#
TSB65	005304	1759	1764#
TST1	001176	460#	
TST10	001520	607#	
TST100	005622	1923#	
TST101	005644	1937#	
TST102	005666	1951#	
TST103	005710	1965#	
TST104	005732	1979#	
TST105	005754	1993#	
TST106	005776	2007#	
TST107	006020	2021#	
TST11	001556	628#	
TST110	006070	2041#	
TST111	006140	2061#	
TST112	006210	2081#	
TST113	006260	2101#	
TST114	006330	2121#	

TST115	006400	2141#
TST116	006450	2161#
TST117	006520	2181#
TST12	001614	649#
TST120	006570	2201#
TST121	006640	2221#
TST122	006710	2241#
TST123	006760	2261#
TST124	007030	2281#
TST125	007100	2301#
TST126	007150	2321#
TST127	007220	2341#
TST13	001652	670#
TST130	007270	2361#
TST131	007340	2381#
TST132	007410	2401#
TST133	007460	2421#
TST134	007530	2441#
TST135	007600	2461#
TST135	007650	2481#
TST137	007720	2501#
TST14	001710	691#
TST140	007770	2521#
TST141	010040	2541#
TST142	010110	2561#
TST143	010160	2581#
TST144	010230	2601#
TST145	010300	2621#
TST146	010350	2641#
TST147	010420	2661#
TST15	001746	712#
TST150	010470	2681#
TST151	010540	2701#
TST152	010610	2721#
TST153	010660	2741#
TST154	010730	2761#
TST155	011000	2781#
TST156	011050	2801#
TST157	011120	2821#
TST16	002004	733#
TST160	011170	2841#
TST161	011240	2861#
TST162	011310	2881#
TST163	011360	2901#
TST17	002042	754#
TST2	001234	481#
TST20	002100	775#
TST21	002136	796#
TST22	002174	817#
TST23	002232	838#
TST24	002270	859#
TST25	002326	880#
TST26	002364	901#
TST27	002422	922#
TST3	001272	502#
TST30	002464	943

947#















H06

MAINDEC-11-DCFPAB-B TEST OF LDFPS, STFPS, SETF, SETD, SETI, SETL, CFCC  
DCFPAB.F11 CROSS REFERENCE TABLE -- PERMANENT SYMBOLS

MACY11 27(732) 10-SEP-76 11:04 PAGE 75

DEFAULT GLOBALS GENERATED: 0

\*DCFPAB,DCFPAB,SEQ/SOL/CRF/DS:ERFZ/EN:ABS=DSKM:DCFPAB.F11  
RUN-TIME: 17 27 4 SECONDS  
RUN-TIME RATIO: 128/SD=3.4  
CORE USED: 18K (24 PAGES)

