

.REM :

IDENTIFICATION

PRODUCT CODE: MAINDEC-11-DCFPA TO DCFPL-B-E
 PRODUCT NAME: FP11 BASIC INSTRUCTION TESTS
 DATE CREATED: 25-APR-72
 MAINTAINER: DIAGNOSTIC GROUP
 AUTHORS: BOB BRAIN & KEV 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	ADD, ADD, SUBF, SUBC
DCFDE-A	CMDF, CMPD
DCFPF-A	MULF, MULD
DCFPG-A	DIVF, DIVD
DCFPH-A	CLRF, CLRD, TSTF, TSTD, ABSF, ABSD, NEGF, NEGD
DCFFA-A	LDCF, LCCDF, STCF, STCDF
DCFFB-A	LDCF, LDCLF, LCCID, LCCID
DCFFC-A	STCFI, STCFI, STCDI, STCDI
DCFFD-A	LDEXP, STEXP, MODF, MODD

FORM 1-65-DCFPA-8

FPII BASIC INSTRUCTION TEST DCFPA - DCFPL
TABLE OF CONTENTS

PAGE 2

CONTENTS

- 1. ABSTRACT
- 2. REQUIREMENTS
 - 2.1 EQUIPMENT
 - 2.2 STORAGE
 - 2.3 PRELIMINARY PROGRAMS
- 3. LOADING PROCEDURE
- 4. STARTING PROCEDURE
 - 4.1 CONTROL SWITCH SETTINGS
 - 4.2 STARTING ADDRESS
 - 4.3 PROGRAM AND/OR OPERATOR ACTION
- 5. OPERATING PROCEDURE
 - 5.1 OPERATIONAL SWITCH SETTINGS
 - 5.2 SUBROUTINE ABSTRACT
- 6. ERRORS
- 7. RESTRICTIONS
- 8. MISCELLANEOUS
 - 8.1 EXECUTION TIME
 - 8.2 STACK POINTER
 - 8.3 POWER FAIL
- 9. PROGRAM DESCRIPTION

FORM 1-65-DCFPA-8

FPII BASIC INSTRUCTION TEST DCFPA - DCFPL
DESCRIPTION

PAGE 3

1. ABSTRACT

THESE PROGRAMS TEST THE FPII 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

POP11/45 STANDARD COMPUTER WITH FPII 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.

DCFPA-S

FP11 BASIC INSTRUCTION TEST DCFPA - DCFPL
DESCRIPTION

PAGE 4

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. AN ERROR OCCURS. THAT TEST WILL BE LOOPEL UPON COMPLETION OF 256 CONSECUTIVE PASSES WITH NO ERRORS OF 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 LB REGISTER

5.2 SUBROUTINE ABSTRACTS

5.2.1 SCOPE

THIS SUBROUTINE CALL IS PLACED BETWEEN EACH SUBTEST IN INSTRUCTION SECTION. IT RECORDS THE STARTING ADDRESS OF EACH SUBTEST AS IT IS BEING ENTERED IN LOCATION LAD. SCOPE LOOP IS REQUESTED. THE CURRENT SUBTEST WILL BE LOOPEL UPON. SW(11) ON A 1 INHIBITS ITERATION OF SUBTEST. 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 IT IS EXECUTED. THE SUBTEST WILL BE LOOPEL UPON. CONSECUTIVE GOOD PASSES ARE COMPLETED IF SW(9) IS ON A 1 TO INHIBIT TYPEOUTS. PUT SW(13) ON A 1.

Vertical text on the left margin, possibly a page number or reference code, appearing as a series of small characters.

1111-DCFPA-8

FP11 BASIC INSTRUCTION TEST DCFPA - DCFPL
DESCRIPTION

PAGE 6

6.2 ERROR RECOVERY
RESTART AT 200

7. RESTRICTIONS
NONE

8. MISCELLANEOUS

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

8.2 STACK POINTER
STACK IS INITIALLY SET TO 600

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

9. PROGRAM DESCRIPTION

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

Vertical column of characters on the left margin, possibly a scan artifact or a specific test identifier.

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 SW7:0 1 - LOOP ON TEST IN SW7:0
9	LOOP ON ERROR
10	0 - BELL ON PASS COMPLETE 1 - BELL ON ERROR
11	INHIBIT ITERATIONS
12	INHIBIT TRACE TRAP
13	INHIBIT ERROR TYPEOLTS
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 --C
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		INTERUPT DISABLE			
14		ERROR FLAG*			

```

000001 .ENABL ABS
177776 N= 1
177570 PS= 177776
177570 SWR= 177570
177570 DISPLAY=SWR
104400 SCOPE= TRAP
104000 HLT= EMT
000004 TYPE= IOT
000207 BELL= 207
000000 FPS= 20
000000 RO= 20
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 ACC= 20
000001 AC1= %1
000002 AC2= %2
000003 AC3= %3
000004 AC4= %4
000005 AC5= %5
100000 SW15= 100000
040000 SW14= 40000
020000 SW13= 20000
010000 SW12= 10000
004000 SW11= 4000
002000 SW10= 2000
001000 SW09= 1000
000400 SW08= 400
170003 LDUB= 170003
170005 STAD= 170005
170007 STGD= 170007
170006 MRS= 170006
170004 LDSC= 170004

000000 .= 0 ;TRAP CATCHER FROM C - 775
000200 .= 200

000200 000167 000622 JMP BEG

000760 000760
000760 170200
000762 170367 000034 FL*ERR: STFPS FPS
000766 000000 STST FEC
000770 000002 HALT
RTI

```

```

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

001026 012706 000600      BEG:  MOV    #600,SP      **: STACK AT 600 **
001032 012737 001054 000004      MOV    #M1120,2#4    : FIND OUT WHICH MACHINE THIS IS
001040 005737 177772      TST    2#177772     : IS PIRQ THERE?
001044 012767 000006 010462      MOV    #6,YESRT    : FUDGE IN RTT IF 11/45
001052 000403      BR

001054 016737 011616 000010 M1120: MOV    FPTADR,2#10  : LOAD THE ILLEGAL INSTRUCTION VECTOR
                                : WITH THE ADDRESS OF THE FPU.
                                : THE FPU WILL HANDLE THE BAD OPCODES
                                : RESET 4

001062 012737 000006 000004 BEGIN: MOV    #6,2#4
001070 012706 000600      MOV    #600,SP
001074 012737 011534 000014      MOV    #YESRT,2#14 : SET TRACE TRAP VECTOR
001102 012777 012374 011574      MOV    #POWDWN,2DOWNVEC
001110 012777 000340 011570      MOV    #340,2DOWNVEC+2
001116 012737 012574 000020      MOV    #.IOT,2#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 000750 011522      MOV    #FLTERR,2FPVECT : LOAD INTERRUPT VECTOR
001156 012777 000340 011516      MOV    #340,2FPVECT+2 : LOCK UP PROCESSOR
001164 005067 177610      CLR    ICNT
001170 005067 011524      CLR    LAC

```



```

*****
: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

```

```

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

```

```

*****
: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

```

```

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

```

```

*****
: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

```

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
:*****


```

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

```

```

001574 104400
001575 012700 000100
001576 170100
001577 022700 000100
001578 001402
001579 005000
001580 104000

001576 170200
001600 010000
001602 022700 000100
001606 001401
001610 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

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

001624 170205
001626 010000
001628 022705 000200
001630 001401
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

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
*****

```

```

001634 104400
001636 012702 000400
001642 170102
001644 022702 000400
001648 001402
001652 005000
001654 104000

001634 170202
001636 010000
001638 022702 000400
001640 001401
001642 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

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

```

001670 170202
001671 010300
001672 022703 000400
001673 001401
001674 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 = MO-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 R2 FPS :SAVE FPS FOR TYPING
CMP #001000,R2 :CHECK FLOATING POINT STATUS
BEQ .+4 :BRANCH IF OK
HLT :FPS NOT EQUAL TO 001000

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

001744 104400
001746 012703 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 = MO-R4

```

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

002024 170204
002026 010400
002030 022704 004000
002034 001401
002036 104000

```

TEST16: SCOPE :SET UP R4
MOV #004000,R4 :LOAD FLOATING POINT STATUS WITH 004000
LDFPS R4 :CHECK R4
CMP #004000,R4 :BRANCH IF OK
BEQ TSA16 :FPS NOT YET STORED
CLR FPS :R4 NOT EQUAL TO 004000
HLT

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 = MD-R3
*****

```

```

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

002062 170203
002064 010300
002066 022703 000000
002072 001401
002074 104000

```

TEST17: SCOPE :SET UP R3
MOV #010000,R3 :LOAD FLOATING POINT STATUS WITH 010000
LDFPS R3 :CHECK R3
CMP #010000,R3 :BRANCH IF OK
BEQ TSA17 :FPS NOT YET STORED
CLR FPS :R3 NOT EQUAL TO 010000
HLT

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 = MD-R2
*****

```

```

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

002118 170202
002120 010200
002122 022702 000000
002124 001401
002126 104000

```

TEST20: SCOPE :SET UP R2
MOV #020000,R2 :LOAD FLOATING POINT STATUS WITH 020000
LDFPS R2 :CHECK R2
CMP #020000,R2 :BRANCH IF OK
BEQ TSA20 :FPS NOT YET STORED
CLR FPS :R2 NOT EQUAL TO 020000
HLT

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
*****

```

```

002154 104400
002154 012701 040000
002154 170101
002154 022701 040000
002154 001402
002154 005000
002154 104000

002156 170201
002160 010100
002162 022701 040000
002166 001401
002170 104000

```

```

TEST21: 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     :R1 NOT EQUAL TO 040000

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     :FPS NOT EQUAL TO 040000

```

```

*****
: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

002214 170205
002216 010500
002220 022705 100000
002224 001401
002226 104000

```

```

TEST22: 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     :R5 NOT EQUAL TO 100000

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     :FPS NOT EQUAL TO 100000

```

```

*****
: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

```

```

TEST23: 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     :R3 NOT EQUAL TO 177756

TSA23:  STFPS  R3      :STORE FLOATING POINT STATUS IN R3
MOV     R3, FPS       :SAVE FPS FOR TYPING
CMP     #177756,R3    :CHECK FLOATING POINT STATUS
BEQ     .+4           :BRANCH IF OK
HLT     :FPS NOT EQUAL TO 177756

```

002255 170203
002256 010300
002257 022703 147756
002258 001401
002259 104300

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 = MO-R2

002266 104400
002270 012702 177755
002274 170102
002276 022702 177755
002302 001402
002304 005300
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 104300

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 = MO-R4

002324 104400
002326 012704 177753
002330 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
002354 022704 147753
002356 001401
002358 104300

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 = MO-R1

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

177747
177747

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 010500
002410 022701
002414 001401
002418 004000

147747

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
002426 170105
002430 022705
002434 001402
002436 005000
002440 104000

177757
177757

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
002450 001401
002454 104000

147757

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

002462 177717

SCOPE
BR TST30 ;BRANCH OVER DATA

002464 012700 002462
002470 170110 002462
002472 022700 002462
002476 001404
002500 010007 170276
002504 005000
002506 104001

170276

DAT30: 177717
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 010007 001004

TSA30: MOV #ANS1, R0 ;MOVE ADDRESS OF ANS1 INTO R0

H02

MACY11-11-00FPA-B
00FPAE.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      *SB30       :BRANCH IF OK
002524 010067 176252    MOV      R0, ANS1     :SAVE R0 FOR TYPING
002530 016700 176252    MOV      ANS2, FPS    :MOVE ANS2 INTO FPS
002534 104001          HLT+1              :R0 NOT EQUAL TO #ANS2

002536 016700 176242    *SB30: 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 JP R0
002560 170100          LDFPS   R0            :LOAD FLOATING POINT STATUS WITH 177657
002562 022700 177657    CMP      #177657, R0   :CHECK R0
002566 001402          BEQ      *SA31       :BRANCH IF OK
002570 005000          CLR     FPS          :FPS NOT YET STORED
002572 104000          HLT                :R0 NOT EQUAL TO 177657

002574 170200          *SA31: STFPS   R0            :STORE FLOATING POINT STATUS IN R0
002576 010000          MOV     R0, FPS      :SAVE FPS FOR TYPING
002600 022700 147657    CMP      #147657, R0   :CHECK FLOATING POINT STATUS
002604 001401          BEQ      .+4         :BRANCH IF OK
002606 104000          HLT                :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     *SA32       :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    *SA32: MOV     #ANS2, R3   :MOVE ADDRESS OF ANS2 INTO R3
002646 170200          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   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     2(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     2(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 #A0B34+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 = M0-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 012200          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   %R6, ANS2      :SAVE R6
003112 170116          LDFPS  (6)           :LOAD FLOATING POINT STATUS WITH 173757
003114 020667 175664  CMP   %R6, ANS2      :DID R6 REMAIN UNCHANGED?
003120 001405          BEQ   TSA36          :BRANCH IF OK
003122 010667 175654  MOV   %R6, ANS1      :SAVE R6 FOR TYPING
003126 016706 175652  MOV   ANS2, %R6      :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   %R6, ANS2      :DID R6 REMAIN UNCHANGED?
003142 001405          BEQ   TSB36          :BRANCH IF OK
003144 010667 175632  MOV   %R6, ANS1      :SAVE R6 FOR TYPING
003150 016706 175630  MOV   ANS2, %R6      :RESTORE R6
003154 104002          HLT+2          :R6 (ANS1) NOT EQUAL TO ANS2

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

```

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		TS37:	SCOPE			
003172	010667	175606		MOV	%6,	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	%6,	ANS2 ;CHECK R6 FOR AUTO-INCREMENT	
003210	001405			BEQ	TSA37	;BRANCH IF OK	
003212	010667	175564		MOV	%6,	ANS1 ;SAVE R6 FOR TYPING	
003216	016706	175562		MOV	ANS2,	%6 ;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	%6,	ANS2 ;CHECK R6 FOR AUTO-DECREMENT	
003240	001405			BEQ	TSA37		;BRANCH IF OK
003242	010667	175534		MOV	%6,	ANS1 ;SAVE R6 FOR TYPING	
003246	016706	175532		MOV	ANS2,	%6 ;RESTORE R6	
003252	104002			HLT+2		;R6 (ANS1) NOT EQUAL TO ANS2	
003254	012600			TSB37:	MOV	(6)+,	FPS ;POP FLOATING POINT STATUS
003256	022730	147757		CMP	#147757,FPS		;CHECK FLOATING POINT STATUS
003262	001401			BEQ	.+4		;BRANCH IF OK
003264	104000			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		TS40:	MOV	%6,	ANS2 ;SAVE R6
003300	012746	003272			MOV	#DAT40,-(6)	;PUSH ADDRESS OF DATA ONTO STACK
003304	170136				LDFPS	(6)+	;LOAD FLOATING POINT STATUS WITH 157757
003306	020667	175472			CMP	%6,	ANS2 ;CHECK R6 FOR AUTO-INCREMENT BY 2
003312	001405				BEQ	TSA40	;BRANCH IF OK
003314	010667	175462			MOV	%6,	ANS1 ;SAVE R6 FOR TYPING
003320	016706	175460			MOV	ANS2,	%6 ;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  0(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, %5   ;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    TSB41      ;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  *TSB41: 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          ACA42: 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          SCOPE
003665 000403          BR      TST44          ;BRANCH OVER DATA

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

003676 012705 003727    TST44: MOV      #ADA44+35,R5      ;MOVE ADDR.+35 OF ADDR. OF DATA INTO R5
003702 170175 177743    LDFPS   @-35(R5)      ;LOAD FLOATING POINT STATUS WITH 052505
003706 022705 003727    CMP     #ADA44+35,R5 ;DID R5 REMAIN UNCHANGED?
003712 001403          BEQ     .+10         ;BRANCH IF OK
003714 010567 175062    MOV     R5,ANS1      ;SAVE R5 FOR TYPING
003720 104001          HLT+1      ;R5 NOT EQUAL TO #ADA44+35

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

003752 022700 042505      CMP     #042505,FPS  ;CHECK FLOATING POINT STATUS
003756 001401          BEQ     .+4         ;BRANCH IF OK
003760 104000          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          SCOPE
003764 170127 146314    TST45: LDFPS   #146314      ;LOAD FLOATING POINT STATUS WITH 146314
003770 000403          BR      .+10        ;BRANCH OVER PC CATCHER
003772 104000          HLT        ;PC FAILURE
003774 104000          HLT        ;PC FAILURE
003776 104000          HLT        ;PC FAILURE
004000 170227          STFPS  (PC,+)      ;STORE FLOATING POINT STATUS IN .+2
004002 000000          ANR45: 0           ;LOCATION FOR ANSWER
004004 016700 177772    MOV     ANR45, FPS   ;MOV STATUS INTO FPS
004010 022700 146314    CMP     #146314,FPS  ;CHECK FLOATING POINT STATUS
004014 001401          BEQ     .+4         ;BRANCH IF OK
004016 104000          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
004024 031443
004026 170137 004024
004028 000403
004030 104000
004032 104000
004034 104000
004036 104000
004038 170237 001000
004040 016700 174700
004042 000401 001700
004044 000401
004046 000401
004048 000401

SCOPE
BR TST46 :BRANCH OVER DATA
DAT46: 031443
TST46: LDFPS DAT46 :LOAD FLOATING POINT STATUS WITH 031443
BR .+10 :BRANCH OVER PC CATCHER
HLT :PC FAILURE
HLT :PC FAILURE
HLT :PC FAILURE
STEPS 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
004066 114611
004070 170167 177770
004072 170267 174700
004074 016700 174676
004076 000401 104611
004078 000401
004080 104000

SCOPE
BR TST47 :BRANCH OVER DATA
DAT47: 114611
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

004082 104400
004084 000403
004086 063146
004088 000403
004090 001000
004092 170177 177770
004094 170277 174700

SCOPE
BR TST50 :BRANCH OVER DATA
DAT50: 063146
ACR50: DAT50
ROR50: ANS1
TST50: LDFPS DAT50 :LOAD FLOATING POINT STATUS WITH 063146
STFPS ROR50 :STORE FLOATING POINT STATUS IN ANS1

004310 104400 174454
004311 000000 040707
004312 000000
004313 000000
004314 143414

TS852: MOV ANS2 FPS :MOVE ANS2 INTO FPS
CMP 0040707,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
004311 000401
004314 143414

SCOPE
BR TS53 :BRANCH OVER DATA
DATA3: 143414

004316 170167 177772
004317 170267 174454
004318 0016700 174450
004319 0020700 143414
004320 0014001
004321 0040000

TS53: LDFPS DATA3 :LOAD FLOATING POINT STATUS WITH 143414
STFPS ANS1 :STORE FLOATING POINT STATUS IN ANS1
MOV ANS1,FPS :MOVE STATUS INTO FPS
CMP 0143414,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

004322 104400 004343
004323 170167 004343
004324 0020700 004343
004325 0020700
004326 0020700
004327 0020700
004328 0020700
004329 0020700
004330 0020700
004331 0020700
004332 0020700
004333 0020700
004334 0020700
004335 0020700
004336 0020700
004337 0020700
004338 0020700
004339 0020700
004340 0020700
004341 0020700
004342 0020700
004343 0020700
004344 0020700
004345 0020700
004346 0020700
004347 0020700
004348 0020700
004349 0020700
004350 0020700
004351 0020700
004352 0020700
004353 0020700
004354 0020700
004355 0020700
004356 0020700
004357 0020700
004358 0020700
004359 0020700
004360 0020700
004361 0020700
004362 0020700
004363 0020700
004364 0020700
004365 0020700
004366 0020700
004367 0020700
004368 0020700
004369 0020700
004370 0020700
004371 0020700
004372 0020700
004373 0020700
004374 0020700
004375 0020700
004376 0020700
004377 0020700
004378 0020700
004379 0020700
004380 0020700
004381 0020700
004382 0020700
004383 0020700
004384 0020700
004385 0020700
004386 0020700
004387 0020700
004388 0020700
004389 0020700
004390 0020700
004391 0020700
004392 0020700
004393 0020700
004394 0020700
004395 0020700
004396 0020700
004397 0020700
004398 0020700
004399 0020700
004400 0020700

TS54: SCOPE
MOV 0034343,R3 :SET UP R3
LDFPS R3 :LOAD FLOATING POINT STATUS WITH 00-343
CMP 0034343,R3 :CHECK R3
BEQ TS854 :BRANCH IF OK
CLR FPS :FPS NOT YET STORED
HLT :R3 NOT EQUAL TO 0034343

004343 0020700
004344 0020700
004345 0020700
004346 0020700
004347 0020700
004348 0020700
004349 0020700
004350 0020700
004351 0020700
004352 0020700
004353 0020700
004354 0020700
004355 0020700
004356 0020700
004357 0020700
004358 0020700
004359 0020700
004360 0020700
004361 0020700
004362 0020700
004363 0020700
004364 0020700
004365 0020700
004366 0020700
004367 0020700
004368 0020700
004369 0020700
004370 0020700
004371 0020700
004372 0020700
004373 0020700
004374 0020700
004375 0020700
004376 0020700
004377 0020700
004378 0020700
004379 0020700
004380 0020700
004381 0020700
004382 0020700
004383 0020700
004384 0020700
004385 0020700
004386 0020700
004387 0020700
004388 0020700
004389 0020700
004390 0020700
004391 0020700
004392 0020700
004393 0020700
004394 0020700
004395 0020700
004396 0020700
004397 0020700
004398 0020700
004399 0020700
004400 0020700

TS854: STFPS R3 :STORE FLOATING POINT STATUS IN R3
MOV R3,FPS :SAVE FPS FOR TYPING
CMP 0034343,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 0020700
004401 0020700
004402 0020700
004403 0020700
004404 0020700
004405 0020700
004406 0020700
004407 0020700
004408 0020700
004409 0020700
004410 0020700
004411 0020700
004412 0020700
004413 0020700
004414 0020700
004415 0020700
004416 0020700
004417 0020700
004418 0020700
004419 0020700
004420 0020700
004421 0020700
004422 0020700
004423 0020700
004424 0020700
004425 0020700
004426 0020700
004427 0020700
004428 0020700
004429 0020700
004430 0020700
004431 0020700
004432 0020700
004433 0020700
004434 0020700
004435 0020700
004436 0020700
004437 0020700
004438 0020700
004439 0020700
004440 0020700
004441 0020700
004442 0020700
004443 0020700
004444 0020700
004445 0020700
004446 0020700
004447 0020700
004448 0020700
004449 0020700
004450 0020700
004451 0020700
004452 0020700
004453 0020700
004454 0020700
004455 0020700
004456 0020700
004457 0020700
004458 0020700
004459 0020700
004460 0020700
004461 0020700
004462 0020700
004463 0020700
004464 0020700
004465 0020700
004466 0020700
004467 0020700
004468 0020700
004469 0020700
004470 0020700
004471 0020700
004472 0020700
004473 0020700
004474 0020700
004475 0020700
004476 0020700
004477 0020700
004478 0020700
004479 0020700
004480 0020700
004481 0020700
004482 0020700
004483 0020700
004484 0020700
004485 0020700
004486 0020700
004487 0020700
004488 0020700
004489 0020700
004490 0020700
004491 0020700
004492 0020700
004493 0020700
004494 0020700
004495 0020700
004496 0020700
004497 0020700
004498 0020700
004499 0020700
004500 0020700

SCOPE
BR TS55 :BRANCH OVER DATA
DATA55: 001616

E03

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

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

004406	012705	004406	TST55:	MOV	#DAT55+2,R5	: MOVE ADDRESS PLUS 2 OF DATA INTO R5
004404	012705	004404		LDFPS	-(R5)	: LOAD FLOATING POINT STATUS WITH 151616
174354	012705	174354		CMP	#DAT55, R5	: DID R5 GET DECREMENTED BY 2?
	012705			BEG	TSASS	: BRANCH IF OK
	012705			MOV	R5,ANS1	: SAVE R5 FOR TYPING
	012705			STFPS	FPS	: STORE FPS FOR TYPING
	012705			HLT+1		: R5 NOT EQUAL TO #DAT55
001006	012705	001006	*SASS:	MOV	#ANS2+2,R5	: MOVE ADDRESS PLUS 2 OF ANS2 INTO R5
001004	012705	001004		STFPS	-(R5)	: STORE FLOATING POINT STATUS WITH 141616
174330	012705	174330		CMP	#ANS2, R5	: DID R5 GET DECREMENTED BY 2?
174326	012705	174326		BEG	TSB55	: BRANCH IF OK
	012705			MOV	R5,ANS1	: SAVE R5 FOR TYPING
	012705			MOV	ANS2, FPS	: MOVE ANS2 INTO FPS
	012705			HLT+1		: R5 NOT EQUAL TO #ANS2
174320	012705	174320	*SB55:	MOV	ANS2, FPS	: MOVE ANS2 INTO FPS
141616	012705	141616		CMP	#141616, FPS	: CHECK FLOATING POINT STATUS
	012705			BEG	TS	: BRANCH IF OK
	012705			HLT+1		: FPS NOT EQUAL TO 141616

 TEST 56: TEST LDFPS (LOAD FLOATING POINT STATUS) AND
 STFPS (STORE FLOATING POINT STATUS).
 FPS = 006141, SRC DST = R1-R4

004500	016141	004500	SCOPE			
	004500		BR	*S56		: BRANCH OVER DATA
004500	012704	004500	*S56:	MOV	#DAT55, R4	: MOVE ADDRESS OF DATA INTO R4
	012704			LDFPS	-(R4)	: LOAD FLOATING POINT STATUS WITH 006141
	012704			CMP	#DAT56, R4	: DID R4 REMAIN UNCHANGED?
	012704			BEG	TSASS	: BRANCH IF OK
	012704			MOV	R4,ANS1	: SAVE R4 FOR TYPING
	012704			STFPS	FPS	: FPS NOT YET STORED
	012704			HLT+1		: R4 NOT EQUAL TO #DAT56
001004	012704	001004	*SAS6:	MOV	#ANS2, R4	: MOVE ADDRESS OF ANS2 INTO R4
001004	012704	001004		STFPS	-(R4)	: STORE FLOATING POINT STATUS WITH 141616
174234	012704	174234		CMP	#ANS2, R4	: DID R4 REMAIN UNCHANGED?
174230	012704	174230		BEG	TSASS	: BRANCH IF OK
	012704			MOV	R4,ANS1	: SAVE R4 FOR TYPING
	012704			MOV	ANS2, FPS	: MOVE ANS2 INTO FPS
	012704			HLT+1		: R4 NOT EQUAL TO #ANS2
174224	012704	174224	*SB56:	MOV	ANS2, FPS	: MOVE ANS2 INTO FPS
006141	012704	006141		CMP	#006141, FPS	: CHECK FLOATING POINT STATUS
	012704			BEG	TS	: BRANCH IF OK
	012704			HLT+1		: FPS NOT EQUAL TO 006141

F03

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

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

MACY11 27(732) 10-SEP-76 11:04 PAGE 31

```

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

```

```

00000000 10400000 170340
00000000 10400000
00000000 10400000
00000000 10400000
00000000 10400000
00000000 10400000
00000000 170227
00000000 00000000
00000000 00000000
00000000 01670000 170340
00000000 02270000 140340
00000000 00140000
00000000 10400000

```

```

TEST57: 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      (R0)+        :STORE FLOATING POINT STATUS IN R0
          ANR57:      :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
*****

```

```

004600 10400000
004600 00040000
004630 007417
004634 001004
004636 004632
004640 012700 004640
004644 170150
004648 022700 004636
004652 001400
004656 010367 174022
004660 170200
004664 104000
004670 012700 004636
004674 170250
004678 022700 004634
004682 001400
004686 010367 174076
004690 016700 174076
004694 104000
004698 00167000 174076
004702 0007417 007417
004706 00140000
004710 10400000

```

```

          SCOPE
          BR         TEST60      :BRANCH OVER DATA
DAT60: 007417
ACB60: ANS2
ADA60: DAT60
TEST60: MOV        #ADA60+2, R3   :MOVE ADDRESS+2 OF ADDRESS OF DATA INTO R3
          LDFPS      @-R3         :LOAD FLOATING POINT STATUS
          CMP        #ACB60, 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        #ADA60+2, R3   :RESET R3
          STFPS      @-R3         :STORE FLOATING POINT STATUS IN ANS2
          CMP        #ACB60, 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 #ACB60
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

```


H03

MANDE:-1-CCPA-B
COPY:1

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

MAY11 27(732) 10-SEP-76 11:04 PAGE 33

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

```

005060 104400          SCOPE
005062 000403          BR      *ST63          :BRANCH OVER DATA

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

005072 012701 005123    *ST63: MOV      #ADA63+35,R1      :MOVE ADDR.+35 OF ADDR. OF DATA INTO R1
005076 170171 177743    LDFPS   0-35(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  076(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      *ST64          :BRANCH OVER DATA

005162 036054          DAT64: 036054

005164 170137 005162    *ST64: LDFPS   00DAT64      :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  0ANS1        :STORE FLOATING POINT STATUS IN ANS1
005204 016700 173672    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			BEG 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	*SA65:	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			BEG TSA65	: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	TSA65:	MOV ANS2,FPS	:MOVE ANS2 INTO FPS
005310	022700	103607		CMP #103607,FPS	:CHECK FLOATING POINT STATUS
005314	001401			BEG .+1	: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	*T66:	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			BEG TSA66	:BRANCH IF OK
005336	005000			CLR FPS	:FPS NOT YET STORED
005340	104000			HLT	:R4 NOT EQUAL TO 074150
005342	170204		*SA66:	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			BEG .+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
005364 170001 LDFPS #000000 :LOAD FLOATING POINT STATUS WITH 000000
005366 170200 SETF :SET FLOATING MODE
005370 022700 000000 STFPS FPS :STORE FLOATING POINT STATUS IN FPS
005374 001401 CMP #000000,FPS :CHECK FLOATING POINT STATUS
005376 104000 BEQ .+4 :BRANCH IF OK
HLT :FPS NOT EQUAL TO 000000

```

```

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

```

```

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

```

```

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

```

```

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

```

```

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

```

```

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

```

```

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

```


K03

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		SCOPE		
005600	170127	000000	TST77: LDFPS	#000000	:LOAD FLOATING POINT STATUS WITH 000000
005604	170002		SETI		:SET INTEGER MODE
005606	170200		STFPS	FPS	:STORE FLOATING POINT STATUS IN FPS
005610	022700	000000	CMP	#000000,FPS	:CHECK FLOATING POINT STATUS
005614	001401		BEQ	+.4	:BRANCH IF OK
005616	104000		HLT		:FPS NOT EQUAL TO 000000

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

005620	104400		SCOPE		
005622	170127	177757	TST100: LDFPS	#177757	:LOAD FLOATING POINT STATUS WITH 177757
005626	170002		SETI		:SET INTEGER MODE
005630	170200		STFPS	FPS	:STORE FLOATING POINT STATUS IN FPS
005632	022700	147657	CMP	#147657,FPS	:CHECK FLOATING POINT STATUS
005636	001401		BEQ	+.4	:BRANCH IF OK
005640	104000		HLT		:FPS NOT EQUAL TO 147657

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

005642	104400		SCOPE		
005644	170127	125252	TST101: LDFPS	#125252	:LOAD FLOATING POINT STATUS WITH 125252
005650	170002		SETI		:SET INTEGER MODE
005652	170200		STFPS	FPS	:STORE FLOATING POINT STATUS IN FPS
005654	022700	105252	CMP	#105252,FPS	:CHECK FLOATING POINT STATUS
005660	001401		BEQ	+.4	:BRANCH IF OK
005662	104000		HLT		:FPS NOT EQUAL TO 105252

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

005664	104400		SCOPE		
005666	170127	052505	TST102: LDFPS	#052505	:LOAD FLOATING POINT STATUS WITH 052505
005672	170002		SETI		:SET INTEGER MODE
005674	170200		STFPS	FPS	:STORE FLOATING POINT STATUS IN FPS
005676	022700	042405	CMP	#042405,FPS	:CHECK FLOATING POINT STATUS
005702	001401		BEQ	+.4	:BRANCH IF OK
005704	104000		HLT		:FPS NOT EQUAL TO 042405

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

M03

MAINDEC-11-DCFPA-B
DCFPAS.P11

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

MACY11 27(732) 10-SEP-76 11:04 PAGE 38

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			SCOPE			
006020	170127	000000		TST107: LDFPS	#000000		:LOAD FLOATING POINT STATUS
006024	170000			CFCC			:COPY FLOATING CONDITION CODES
006026	013767	177776	172746	MOV	2#PS,ANSI		:GET CPU STATUS
006034	042767	177760	172740	BIC	#177760,ANSI		:SAVE CONDITION CODES
006042	170200			STFPS	FPS		:STORE FLOATING POINT STATUS IN FPS
006044	022700	000000		CMP	#000000,FPS		:CHECK FLOATING POINT STATUS
006050	001401			BEQ	+.4		:BRANCH IF OK
006052	104000			HLT			:FPS NOT EQUAL TO 000000
006054	022767	000000	172720	CMP	#00, ANSI		:CHECK CONDITION CODES
006062	001401			BEQ	+.4		:BRANCH IF OK
006064	104000			HLT			:CONDITION CODES NOT EQUAL TO 00

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

006066	104400			SCOPE			
006070	170127	177757		TST110: LDFPS	#177757		:LOAD FLOATING POINT STATUS
006074	170000			CFCC			:COPY FLOATING CONDITION CODES
006076	013767	177776	172676	MOV	2#PS,ANSI		:GET CPU STATUS
006104	042767	177760	172670	BIC	#177760,ANSI		:SAVE CONDITION CODES
006112	170200			STFPS	FPS		:STORE FLOATING POINT STATUS IN FPS
006114	022700	147757		CMP	#147757,FPS		:CHECK FLOATING POINT STATUS
006120	001401			BEQ	+.4		:BRANCH IF OK
006122	104000			HLT			:FPS NOT EQUAL TO 147757
006124	022767	000017	172650	CMP	#17, ANSI		:CHECK CONDITION CODES
006132	001401			BEQ	+.4		:BRANCH IF OK
006134	104000			HLT			:CONDITION CODES NOT EQUAL TO 17

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

006136	104400			SCOPE			
006140	170127	000001		TST111: LDFPS	#000001		:LOAD FLOATING POINT STATUS
006144	170000			CFCC			:COPY FLOATING CONDITION CODES
006146	013767	177776	172626	MOV	2#PS,ANSI		:GET CPU STATUS
006154	042767	177760	172620	BIC	#177760,ANSI		:SAVE CONDITION CODES
006162	170200			STFPS	FPS		:STORE FLOATING POINT STATUS IN FPS
006164	022700	000001		CMP	#000001,FPS		:CHECK FLOATING POINT STATUS
006170	001401			BEQ	+.4		:BRANCH IF OK
006172	104000			HLT			:FPS NOT EQUAL TO 000001
006174	022767	000001	172600	CMP	#01, ANSI		:CHECK CONDITION CODES
006202	001401			BEQ	+.4		:BRANCH IF OK
006204	104000			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
LDFPS #000002 :LOAD FLOATING POINT STATUS
CFCC :COPY FLOATING CONDITION CODES
MOV #0FPS,ANS1 :GET CPU STATUS
BIC #177760,ANS1 :SAVE CONDITION CODES
STFPS FPS :STORE FLOATING POINT STATUS IN FPS
CMP #000002,FPS :CHECK FLOATING POINT STATUS
BEQ .+4 :BRANCH IF OK
HLT :FPS NOT EQUAL TO 000002

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
LDFPS #000004 :LOAD FLOATING POINT STATUS
CFCC :COPY FLOATING CONDITION CODES
MOV #0FPS,ANS1 :GET CPU STATUS
BIC #177760,ANS1 :SAVE CONDITION CODES
STFPS FPS :STORE FLOATING POINT STATUS IN FPS
CMP #000004,FPS :CHECK FLOATING POINT STATUS
BEQ .+4 :BRANCH IF OK
HLT :FPS NOT EQUAL TO 000004

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
LDFPS #000010 :LOAD FLOATING POINT STATUS
CFCC :COPY FLOATING CONDITION CODES
MOV #0FPS,ANS1 :GET CPU STATUS
BIC #177760,ANS1 :SAVE CONDITION CODES
STFPS FPS :STORE FLOATING POINT STATUS IN FPS
CMP #000010,FPS :CHECK FLOATING POINT STATUS
BEQ .+4 :BRANCH IF OK
HLT :FPS NOT EQUAL TO 000010

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
006404 170000
006406 013767
006414 042767
006422 170200
006424 022700
006430 001401
006432 104000

000000
177776 172366
177760 172360
000000

TST115: SCOPE :
LDFPS #000000 :LOAD FLOATING POINT STATUS
CFCC :COPY FLOATING CONDITION CODES
MOV #0PS,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

006434 022767
006442 001401
006444 104000

000000 172340

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
006454 170000
006456 013767
006464 042767
006472 170200
006474 022700
006500 001401
006502 104000

000040
177776 172316
177760 172310
000040

TST116: SCOPE :
LDFPS #000040 :LOAD FLOATING POINT STATUS
CFCC :COPY FLOATING CONDITION CODES
MOV #0PS,ANS1 :GET CPU STATUS
BIC #177760,ANS1 :SAVE CONDITION CODES
STFPS FPS :STORE FLOATING POINT STATUS IN FPS
CMP #000040,FPS :CHECK FLOATING POINT STATUS
BEQ .+4 :BRANCH IF OK
HLT :FPS NOT EQUAL TO 000040

006504 022767
006512 001401
006514 104000

000000 172270

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
006524 170000
006526 013767
006534 042767
006542 170200
006544 022700
006550 001401

000100
177776 172246
177760 172240
000100

TST117: SCOPE :
LDFPS #000100 :LOAD FLOATING POINT STATUS
CFCC :COPY FLOATING CONDITION CODES
MOV #0PS,ANS1 :GET CPU STATUS
BIC #177760,ANS1 :SAVE CONDITION CODES
STFPS FPS :STORE FLOATING POINT STATUS IN FPS
CMP #000100,FPS :CHECK FLOATING POINT STATUS
BEQ .+4 :BRANCH IF OK


```

006552 104000          HLT          :FPS NOT EQUAL TO 000100
006554 022767 000000 172220      CMP      #00.   ANS1   :CHECK CONDITION CODES
006556 001401          BEQ      .+4         :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      TEST120: LDFPS   #000200      :LOAD FLOATING POINT STATUS
006574 170000          CFCC      :COPY FLOATING CONDITION CODES
006576 013767 177776 172176      MOV      #FPS,ANS1   :GET CPU STATUS
006604 042767 177760 172170      BIC     #177760,ANS1 :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      .+4         :BRANCH IF OK
006622 104000          HLT          :FPS NOT EQUAL TO 000200

006624 022767 000000 172150      CMP      #00.   ANS1   :CHECK CONDITION CODES
006626 001401          BEQ      .+4         :BRANCH IF OK
006628 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      TEST121: LDFPS   #000400      :LOAD FLOATING POINT STATUS
006644 170000          CFCC      :COPY FLOATING CONDITION CODES
006646 013767 177776 172126      MOV      #FPS,ANS1   :GET CPU STATUS
006654 042767 177760 172120      BIC     #177760,ANS1 :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      .+4         :BRANCH IF OK
006672 104000          HLT          :FPS NOT EQUAL TO 000400

006674 022767 000000 172100      CMP      #00.   ANS1   :CHECK CONDITION CODES
006702 001401          BEQ      .+4         :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      TEST122: LDFPS   #001000      :LOAD FLOATING POINT STATUS
006714 170000          CFCC      :COPY FLOATING CONDITION CODES
006716 013767 177776 172056      MOV      #FPS,ANS1   :GET CPU STATUS

```



```

007076 104400
007100 170127 010000 *T125: SCOPE
007104 170000 LDFPS #010000 :LOAD FLOATING POINT STATUS
007106 013767 177776 171666 CFCC :COPY FLOATING CONDITION CODES
007114 042767 177760 171660 MOV #FPS,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 *T126: SCOPE
007154 170000 LDFPS #020000 :LOAD FLOATING POINT STATUS
007156 013767 177776 171616 CFCC :COPY FLOATING CONDITION CODES
007164 042767 177760 171610 MOV #FPS,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 *T127: SCOPE
007224 170000 LDFPS #040000 :LOAD FLOATING POINT STATUS
007226 013767 177776 171546 CFCC :COPY FLOATING CONDITION CODES
007234 042767 177760 171540 MOV #FPS,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
:*****

007326 104400
007327 170127 100000
0073274 170000
0073276 013767 177776 171475
0073304 042767 177760 171470
0073312 170200
0073314 022700 100000
0073320 001401
0073322 104000

0073324 022767 000000 171450
0073332 001401
0073334 104000

TST130: SCOPE
LDFPS #100000 :LOAD FLOATING POINT STATUS
CFCC :COPY FLOATING CONDITION CODES
MOV @FPS,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 @FPS,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 @FPS,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

TEST133: SCOPE #177753 :LOAD FLOATING POINT STATUS
LDFPS :COPY FLOATING CONDITION CODES
CFCC :GET CPU STATUS
MOV 2#PS,ANS1 :SAVE CONDITION CODES
BIC #177760,ANS1 :STORE FLOATING POINT STATUS IN FPS
STFPS FPS :CHECK FLOATING POINT STATUS
CMP #147753,FPS :BRANCH IF OK
BEQ .+4 :FPS NOT EQUAL TO 147753
HLT

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

TEST134: SCOPE #177747 :LOAD FLOATING POINT STATUS
LDFPS :COPY FLOATING CONDITION CODES
CFCC :GET CPU STATUS
MOV 2#PS,ANS1 :SAVE CONDITION CODES
BIC #177760,ANS1 :STORE FLOATING POINT STATUS IN FPS
STFPS FPS :CHECK FLOATING POINT STATUS
CMP #147747,FPS :BRANCH IF OK
BEQ .+4 :FPS NOT EQUAL TO 147747
HLT

007564 022767 000007 171210
007570 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
007634 104000

TEST135: SCOPE #177757 :LOAD FLOATING POINT STATUS
LDFPS :COPY FLOATING CONDITION CODES
CFCC :GET CPU STATUS
MOV 2#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

```

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          @FPS,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          @FPS,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          @FPS,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 JK
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 SCOPE
0111120 170127 107050 TST157: LDFPS #107050 :LOAD FLOATING POINT STATUS
0111124 170000 CFCC :COPY FLOATING CONDITION CODES
0111126 013767 177776 167646 MOV #0PS,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 SCOPE
0111170 170127 070707 TST160: LDFPS #070707 :LOAD FLOATING POINT STATUS
0111174 170000 CFCC :COPY FLOATING CONDITION CODES
0111176 013767 177776 167576 MOV #0PS,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 MOV #0PS,ANSI :COPY FLOATING CONDITION CODES
011254 042767 177760 167520 BIC #177760,ANSI :GET CPU STATUS
011262 170200 STFPS FPS :SAVE CONDITION CODES
011264 022700 143414 CMP #143414,FPS :STORE FLOATING POINT STATUS IN FPS
011270 001401 BEQ .+4 :CHECK FLOATING POINT STATUS
011272 104000 HLT :BRANCH IF OK
:FPS NOT EQUAL TO 143414

011274 022767 000014 167500 CMP #14, ANSI :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 MOV #0PS,ANSI :COPY FLOATING CONDITION CODES
011324 042767 177760 167450 BIC #177760,ANSI :GET CPU STATUS
011332 170200 STFPS FPS :SAVE CONDITION CODES
011334 022700 004343 CMP #004343,FPS :STORE FLOATING POINT STATUS IN FPS
011340 001401 BEQ .+4 :CHECK FLOATING POINT STATUS
011342 104000 HLT :BRANCH IF OK
:FPS NOT EQUAL TO 034343

011344 022767 000003 167430 CMP #03, ANSI :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, CONCITION CODES = 16
*****

```

```

011356 104400
011360 170127 161616 TST163: SCOPE
011364 170000 LDFPS #161616 :LOAD FLOATING POINT STATUS
011366 013767 177776 167406 MOV #0PS,ANSI :COPY FLOATING CONDITION CODES
011374 042767 177760 167400 BIC #177760,ANSI :GET CPU STATUS
011402 170200 STFPS FPS :SAVE CONDITION CODES
011404 022700 141616 CMP #141616,FPS :STORE FLOATING POINT STATUS IN FPS
011410 001401 BEQ .+4 :CHECK FLOATING POINT STATUS
011412 104000 HLT :BRANCH IF OK
:FPS NOT EQUAL TO 141616

011414 022767 000016 167360 CMP #16, ANSI :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	IS	:NO!
011440	012767	000207	001246		MOV	#BELL,TYPE	:TYPE A BELL
011446	000004	012714			TYPE	..TYPE	
011452	005046			IS:	CLR	-(6)	:CLEAR TRACE TRAP
011454	032737	010000	177570		BIT	#SW12,2#SWR	:RUN WITH TRT?
011462	001010				BNE	2S	
011464	005167	001226			COM	TRPB	
011470	100005				BPL	2S	
011472	052716	000020			BIS	#20,(6)	:SET TRACE TRAP
011476	012746	011530			MOV	#3S,-(6)	:JUMP TO START OF TEST
011502	000002				RTI		
011504	012746	011512		2S:	MOV	#4S,-(6)	:JUMP TO START OF TEST
011510	000002				RTI		
011512	013700	000042		4S:	MOV	#42,RC	:GET MONITOR ADDRESS
011516	001404				BEG	2S	:IF NONE
011520	004710				JSP	..(2)	:GO TO MONITOR
011522	000240				NOP		
011524	000240				NOP		
011526	000240				NOP		
011530	000137	000200		3S:	JMP	#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	IS	
011546	123767	177570	167224		CMPB	#SWR,ICNT	:ON RIGHT TEST? *SWT-C*
011554	001437				BEG	OVER	
011556	113703	177570		IS:	MOVB	#SWR,R3	:GET LB BITS
011562	170003				LDLB		
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	2S	:BRANCH IF FIRST
011612	126767	001106	167161		CMPB	TIMES,ICNT+1	:DONE?
011620	001013				BNE	KIT	:BRANCH IF NOT
011622	112767	000001	167151	2S:	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,#DISP	:DISPLAY TEST NO. AND ITERATION COUNT
011646	000002				RTI		:RETURN
011650	105267	167125		3S:	INCB	ICNT+1	
011654	016737	167120	177570	OVER:	MOV	ICNT,#DISP	:SET UP DISPLAY
011662	005767	001032			LAD		:FIRST ONE?
011666	001760				BEG	SAVLAD	
011670	016737	001024			LAD	..E	:FLDGE RETURN ADDRESS
011674	000002				RTI		:RETURN

EOS

MACY11-27-732-10-SEP-76 11:04 PAGE 56

TEST OF DFPS, STFPS, SETF, SETD, SETI, SETL, CFCC
 HL ROUTINE (ERROR TYPEOUT)

MACY11 27732 10-SEP-76 11:04 PAGE 56

Address	Instruction	OpCode	Operand 1	Operand 2	Comment
011676	032737	002000	177570	.TRP:	BIT #SW10,2#SWR :BELL ON ERROR
011704	001405				:NO - SKIP
011706	012767	000207	001000		MOV #BELL,TYPE :TYPE A BELL
011714	000004	012714			TYPE
011720	004767	000406		18:	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				MOV #2(6)+,R4 :AMOUNT OF DATA IN TABLE
012004	001426				BEG 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
012044	000004	012671		28:	TYPE SPACE+4
012050	012305				MOV (3)+,TTY
012052	004767	000114			JSR %7,PRINTR :TYPE LEADING ZERO'S
012056	005304				DEC R4
012060	001371				BNE 2\$
012062	005700			38:	ST FPS
012064	100016				BPL 4\$
012066	000004	012665			TYPE SPACE
012072	170367	166724			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			48:	MOV (6)+,R4
012124	005737	177570			TST 2#SWR
012130	100001				BPL .+4
012132	000000				HALT
012134	032737	001000	177570		BIT #SW09,2#SWR :CHECK FOR INHIBIT LOOP ON ERROR
012142	001001				BNE .+4 :SKIP IF CONTINUE
012144	000002				RTY
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			MOV 2#SWR,R3 :PUT MICROBREAK ADDRESS IN R3
012164	170000				LOADR :LOAD MICROBREAK
012166	000627				BR :LOOP ON TEST UNTIL NO ERRORS

```

012172 112767 000001 165600 PRINTR: MOV3      #1,45          ;SET ZERO FILL SWITCH
012174 000402          BR          +6
012176 005067 000122          PRINTS: CLR      45          ;SUPPRESS LEADING ZERO'S
012178 112767 177772 000115 MOV3      #-6,45+1      ;SET COUNT
012180 010446          MOV      R4, -(6)      ;SAVE R4
012182 012704 012320          MOV      #35,R4       ;SET POINTER TO FIRST ASCII CHAR.
012184 105014          CLR3     (4)          ;CLEAR FIRST BYTE
012186 000402          BR          25        ;ROTATE FIRST BIT
012188 105014          CLR3     (4)          ;CLEAR BYTE OF CHARACTER
012190 006105          ROL      TTY          ;ROTATE BIT INTO C
012192 106114          ROL3     (4)          ;PACK IT
012194 006105          ROL      TTY          ;ROTATE BIT INTO C
012196 106114          ROL3     (4)          ;PACK IT
012198 006105          ROL      TTY          ;ROTATE BIT INTO C
012200 106114          ROL3     (4)          ;PACK IT
012202 012242          TSTB     (4)
012204 105714          BEQ      +6
012206 001402          INCB     45+6
012208 105267 000054          INCB     45
012210 105767 000050          TSTB     45
012212 001402          BEQ      +6
012214 152724 000060          B15B     #0,(4)+      ;CHECK FILL SWITCH
012216 105267 000037          INCB     45+1
012218 001355          BNE      15
012220 022704 012320          CMP      #35,R4
012222 001002          BNE      +6
012224 112724 000060          MOV3     #0,(4)+      ;MAKE INTO ASCII CHAR
012226 105014          CLR3     (4)
012228 000004 012320          TYPE     135
012230 012604          MOV      (6)+,R4
012232 000207          RTS
012320 000004          35: .BLKW 4
012330 000000          45: 0

012332 005267 000364          ERROR: INC      ERRORS      ;COUNT ERRORS
012334 132737 000001 000041 BYTE     #1,2041      ;AUTO MODE?
012336 001412          BEQ      15
012338 022767 000010 000346          CMP      #10,ERRORS   ;NO?
012340 001006          BNE      15
012342 013700 000042          MOV      #42,R0       ;TOO MANY?
012344 001402          BEQ      15
012346 005037 000042          MOV      #42,R0       ;NOT YET
012348 004711          CLR      0
012350 000207          RTS                    ;GET ADDRESS
                                ;FORGET IT IF ZERO
                                ;ZAP 42
                                ;CALL THE MONITOR
                                ;RETURN

```

```

012570 003306 POWDWN: MOV #ILLUP, @UPVEC :SET FOR FAST JP
00C340 00C302 MOV #340, @UPVEC+2 :PRIO:7
STFPS - (6) :GET THE FPS
SETD :SAVE AC'S
STD ACC, - (6)
STD AC1, - (6)
STD AC2, - (6)
STD AC3, - (6)
STD AC4, ACC
STD AC5, ACC
STD ACC, - (6)
MOV @R0, - (6) :SAVE REGISTERS
MOV @R1, - (6)
MOV @R2, - (6)
MOV @R3, - (6)
MOV @R4, - (6)
MOV @R5, - (6)
000220 000226 SP, SAVES :SAVE SP
012450 010667 MOV @POWUP, @UPVEC :SET UP VECTOR
012454 0:2777
012462 000000 HALT

012464 016706 POWUP: MOV SAVES, SP :GET SP
012470 005001 CLR R1 :WAIT LOOP FOR THE TTY
012472 00520: R1 :S:
012474 001376 BRZ, R1, #0 :GET THE REGISTERS
MOV @R0, - (6)
MOV @R1, - (6)
MOV @R2, - (6)
MOV @R3, - (6)
MOV @R4, - (6)
MOV @R5, - (6)
012500 012604
012502 012603
012504 012602
012506 012601
012510 012600
012512 170011 SETD
012514 172426 STD
012516 174004 STD
012518 172426 STD
012520 174004 STD
012522 172726 LDD
012524 172626 LDD
012526 172526 LDD
012528 172426 LDD
012530 170126 LDDFPS
012532 012777 MOV @POWDOWN, @DOWNVEC :RESTORE FPS
012534 000140 MOV #340, @DOWNVEC+2 :SET UP THE POWER DOWN VECTOR
012536 000134 TYPE :ASCIZ '15:12 "POWER"
012538 000002 RTN

000000 :THE POWER UP SEQUENCE WAS STARTED
000000 :BEFORE THE POWER DOWN AND COMPLETED

```

H05

MACRO-11 - CCFPA-3
CONFIDENTIAL

TEST OF JOBS, STOPS, SETF, SETD, SETI, SETL, CFCC
TYPE ROUTINE AND DATA AREA

MAC-11 27'732 10-SEP-75 11:04 PAGE 59

012674	010546			.NOT:	MOV	TTY, -(6)	:SAVE TTY
012676	017605	000002		1\$:	MOV	2(6), TTY	:GET ADDRESS TO BE TYPED
012602	105715				TSTB	(TTY)	:TERMINATOR?
012604	001406				BEG	2\$	
012606	112537	177566			MOV	(TTY)+, 2#177566	:LOAD AND TYPE THE CHARACTER
012612	105737	177564			TSTB	2#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				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		DNVVEC:	24, 26		:POWER DOWN VECTOR ADDRESS
012714	000024	000026		UPVEC:	24, 26		:POWER UP VECTOR ADDRESS
012716	000000			.TYPE:	000000		
012720	000000			TRPB:	000000		
012722	000000			LAD:	000000		:LOOP ADDRESS
012724	000000			ERRORS:	000000		:ERROR COUNT
	000377			FINES:	000000		:ITERATION COUNT
	000000			.END			

K05

MAINDEC-11-DCFPA-B
DCFPAB.F11

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

MACY11 27'732) 10-SEP-76 11:04 PAGE 53

		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											
		416*	450*	2941	2949	2951	2953*	2954*	2956	2959*	2960	3013*		
		3065	3112*											
		2946	2952	2959*	3015	3018								
		451*	2955*	2961	2963	3144*								
		401*												
		397*												
		400*												
		429	434*											
		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*	956*
		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*											
		382*	1339*	1584*	2970*	3049*	3062*	3063*						
		440	3065*	3107										
		3084	3087*											
		2978	2981	2994	3002	3005	3020*							
		3022*												
		366*	2023	2043	2063	2083	2103	2123	2143	2163	2183	2203	2223	2243
		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*											
		374*	443*	460*	461	462	467*	468	469	628*	629	630	635*	636
		637	712*	713	714	719*	720	721	947*	948	949	950	955*	956*
		957	959	976*	977	978	983*	984	985	1444*	1445	1446	1449	1452*
		1453*	1454	1456	1748*	1749	1750	1752	1756*	1757*	1758	1760	2930*	2980
		3059*	3077	3096*										
		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	1290*	1281*	1282	1284
		1694*	1695	1696	1698	1701*	1702*	1704	1706	3078	3088*	3089*	3095*	
		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*					
		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*	2993*	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	525	530#
TSA40	003326	1174	1179#
TSA41	003422	1207	1212#
TSA42	003526	1242	1248#
TSA43	003620	1275	1280#
TSA5	001405	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#
TSA830	002536	958	963#
TSA832	002670	1012	1017#
TSA834	003026	1068	1073#
TSA836	003156	1118	1123#
TSA837	003254	1148	1153#
TSA840	003360	1183	1188#
TSA941	003450	1215	1220#
TSA942	003550	1250	1255#
TSA943	003650	1283	1288#
TSA952	004274	1455	1460#
TSA955	004460	1528	1533#
TSA956	004554	1561	1566#
TSA960	004712	1616	1621#
TSA962	005044	1670	1675#
TSA965	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	2451#
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#

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

TEST OF	DFPS	STFPS	SETF	SETD	SETI	SETL	CFCC							
002554	976													
002676	997	1001												
002706	1030													
002754	1051	1057												
003044	1086													
003102	1107													
003172	1136													
003332	523													
003376	1166	1170												
003476	1200	1204												
003476	1233	1239												
003572	1268	1272												
003676	1301	1307												
003764	1334													
004026	1354	1358												
004070	1377	1381												
001366	544													
004126	1396	1402												
004154	1417													
004222	1428	1444												
004216	1473	1477												
004244	1492													
004406	1513	1517												
004502	1546	1550												
004572	1579													
001424	565													
004640	1599	1605												
004740	1634	1640												
004772	1655	1659												
005072	1688	1694												
005164	1721	1725												
005226	1744	1748												
005322	1777													
005360	1797													
001462	586													
005402	1811													
005424	1825													
005446	1839													
005470	1853													
005512	1867													
005534	1881													
005556	1895													
005600	1909													
000005	380	2977*	2980*	2993*	3001*	3004*	3029*	3031*	3033*	3115	3115*	3117	3119	
000004	3127*	3128*	3129	3130*										
0012710	371	2919	2969	2974	2979	2982	2992	2999	3003	3047	3109			
0010004	3065*	3066*	3084*	3141										
0010004	421*	429*	2927*											
0010004	403*	404	405*	409	415	470	491	512	533	554	575	596	617	638
0010004	638	659	680	701	722	743	764	785	806	827	848	869	890	911
0010004	911	932	953	986	1019	1050	1081	1112	1143	1174	1205	1236	1267	1298
0010004	1298	1310	1318	1323	1325	1327	1329	1331	1333	1335	1337	1339	1341	1343
0010004	1343	1345	1347	1349	1351	1353	1355	1357	1359	1361	1363	1365	1367	1369
0010004	1369	1371	1373	1375	1377	1379	1381	1383	1385	1387	1389	1391	1393	1395
0010004	1395	1397	1399	1401	1403	1405	1407	1409	1411	1413	1415	1417	1419	1421
0010004	1421	1423	1425	1427	1429	1431	1433	1435	1437	1439	1441	1443	1445	1447
0010004	1447	1449	1451	1453	1455	1457	1459	1461	1463	1465	1467	1469	1471	1473
0010004	1473	1475	1477	1479	1481	1483	1485	1487	1489	1491	1493	1495	1497	1499
0010004	1499	1501	1503	1505	1507	1509	1511	1513	1515	1517	1519	1521	1523	1525
0010004	1525	1527	1529	1531	1533	1535	1537	1539	1541	1543	1545	1547	1549	1551
0010004	1551	1553	1555	1557	1559	1561	1563	1565	1567	1569	1571	1573	1575	1577
0010004	1577	1579	1581	1583	1585	1587	1589	1591	1593	1595	1597	1599	1601	1603
0010004	1603	1605	1607	1609	1611	1613	1615	1617	1619	1621	1623	1625	1627	1629
0010004	1629	1631	1633	1635	1637	1639	1641	1643	1645	1647	1649	1651	1653	1655
0010004	1655	1657	1659	1661	1663	1665	1667	1669	1671	1673	1675	1677	1679	1681
0010004	1681	1683	1685	1687	1689	1691	1693	1695	1697	1699	1701	1703	1705	1707
0010004	1707	1709	1711	1713	1715	1717	1719	1721	1723	1725	1727	1729	1731	1733
0010004	1733	1735	1737	1739	1741	1743	1745	1747	1749	1751	1753	1755	1757	1759
0010004	1759	1761	1763	1765	1767	1769	1771	1773	1775	1777	1779	1781	1783	1785
0010004	1785	1787	1789	1791	1793	1795	1797	1799	1801	1803	1805	1807	1809	1811
0010004	1811	1813	1815	1817	1819	1821	1823	1825	1827	1829	1831	1833	1835	1837
0010004	1837	1839	1841	1843	1845	1847	1849	1851	1853	1855	1857	1859	1861	1863
0010004	1863	1865	1867	1869	1871	1873	1875	1877	1879	1881	1883	1885	1887	1889
0010004	1889	1891	1893	1895	1897	1899	1901	1903	1905	1907	1909	1911	1913	1915
0010004	1915	1917	1919	1921	1923	1925	1927	1929	1931	1933	1935	1937	1939	1941
0010004	1941	1943	1945	1947	1949	1951	1953	1955	1957	1959	1961	1963	1965	1967
0010004	1967	1969	1971	1973	1975	1977	1979	1981	1983	1985	1987	1989	1991	1993
0010004	1993	1995	1997	1999	2001	2003	2005	2007	2009	2011	2013	2015	2017	2019
0010004	2019	2021	2023	2025	2027	2029	2031	2033	2035	2037	2039	2041	2043	2045
0010004	2045	2047	2049	2051	2053	2055	2057	2059	2061	2063	2065	2067	2069	2071
0010004	2071	2073	2075	2077	2079	2081	2083	2085	2087	2089	2091	2093	2095	2097
0010004	2097	2099	2101	2103	2105	2107	2109	2111	2113	2115	2117	2119	2121	2123
0010004	2123	2125	2127	2129	2131	2133	2135	2137	2139	2141	2143	2145	2147	2149
0010004	2149	2151	2153	2155	2157	2159	2161	2163	2165	2167	2169	2171	2173	2175
0010004	2175	2177	2179	2181	2183	2185	2187	2189	2191	2193	2195	2197	2199	2201
0010004	2201	2203	2205	2207	2209	2211	2213	2215	2217	2219	2221	2223	2225	2227
0010004	2227	2229	2231	2233	2235	2237	2239	2241	2243	2245	2247	2249	2251	2253
0010004	2253	2255	2257	2259	2261	2263	2265	2267	2269	2271	2273	2275	2277	2279
0010004	2279	2281	2283	2285	2287	2289	2291	2293	2295	2297	2299	2301	2303	2305
0010004	2305	2307	2309	2311	2313	2315	2317	2319	2321	2323	2325	2327	2329	2331
0010004	2331	2333	2335	2337	2339	2341	2343	2345	2347	2349	2351	2353	2355	2357
0010004	2357	2359	2361	2363	2365	2367	2369	2371	2373	2375	2377	2379	2381	2383
0010004	2383	2385	2387	2389	2391	2393	2395	2397	2399	2401	2403	2405	2407	2409
0010004	2409	2411	2413	2415	2417	2419	2421	2423	2425	2427	2429	2431	2433	2435
0010004	2435	2437	2439	2441	2443	2445	2447	2449	2451	2453	2455	2457	2459	2461
0010004	2461	2463	2465	2467	2469	2471	2473	2475	2477	2479	2481	2483	2485	2487
0010004	2487	2489	2491	2493	2495	2497	2499	2501	2503	2505	2507	2509	2511	2513
0010004	2513	2515	2517	2519	2521	2523	2525	2527	2529	2531	2533	2535	2537	2539
0010004	2539	2541	2543	2545	2547	2549	2551	2553	2555	2557	2559	2561	2563	2565
0010004	2565	2567	2569	2571	2573	2575	2577	2579	2581	2583	2585	2587	2589	2591
0010004	2591	2593	2595	2597	2599	2601	2603	2605	2607	2609	2611	2613	2615	2617
0010004	2617	2619	2621	2623	2625	2627	2629	2631	2633	2635	2637	2639	2641	2643
0010004	2643	2645	2647	2649	2651	2653	2655	2657	2659	2661	2663	2665	2667	2669
0010004	2669	2671	2673	2675	2677	2679	2681	2683	2685	2687	2689	2691	2693	2695
0010004	2695	2697	2699	2701	2703	2705	2707	2709	2711	2713	2715	2717	2719	2721
001000														

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

CROSS
TEST OF
REFERENCE

554	715	869	1033	1189	1385	1568	1766	1969	2122	2297	2484	2682	2891	3111	3342	3584	3837	4091	4356	4631	4916	5211	5516	5831	6156	6491	6836	7191	7556	7931	8316	8711	9116	9531	9956																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																			
1033	1189	1385	1568	1766	1969	2122	2297	2484	2682	2891	3111	3342	3584	3837	4091	4356	4631	4916	5211	5516	5831	6156	6491	6836	7191	7556	7931	8316	8711	9116	9531	9956	10391	10836	11291	11756	12231	12716	13211	13716	14231	14756	15291	15836	16391	16956	17531	18116	18711	19316	19931	20556	21191	21836	22491	23156	23831	24516	25211	25916	26631	27356	28091	28836	29591	30356	31131	31916	32711	33516	34331	35156	35991	36836	37691	38556	39431	40316	41211	42116	43031	43956	44891	45836	46791	47756	48731	49716	50711	51716	52731	53756	54791	55836	56891	57956	59031	60116	61211	62316	63431	64556	65691	66836	68091	69356	70631	71916	73211	74516	75831	77156	78491	79836	81191	82556	83931	85316	86711	88116	89531	90956	92391	93836	95291	96756	98231	99716	101211	102716	104231	105756	107291	108836	110391	111956	113531	115116	116711	118316	119931	121556	123191	124836	126491	128156	129831	131516	133211	134916	136631	138356	140091	141836	143591	145356	147131	148916	150711	152516	154331	156156	157991	159836	161691	163556	165431	167316	169211	171116	173031	174956	176891	178836	180791	182756	184731	186716	188711	190716	192731	194756	196791	198836	200891	202956	205031	207116	209211	211316	213431	215556	217691	219836	221991	224156	226331	228516	230711	232916	235131	237356	239591	241836	244091	246356	248631	250916	253211	255516	257831	260156	262491	264836	267191	269556	271931	274316	276711	279116	281531	283956	286391	288836	291291	293756	296231	298716	301211	303716	306231	308756	311291	313836	316391	318956	321531	324116	326711	329316	331931	334556	337191	339836	342491	345156	347831	350516	353211	355916	358631	361356	364091	366836	369591	372356	375131	377916	380711	383516	386331	389156	391991	394836	397691	400556	403431	406316	409211	412116	415031	417956	420891	423836	426791	429756	432731	435716	438711	441716	444731	447756	450791	453836	456891	459956	463031	466116	469211	472316	475431	478556	481691	484836	487991	491156	494331	497516	500711	503916	507131	510356	513591	516836	520091	523356	526631	529916	533211	536516	539831	543156	546491	549836	553191	556556	559931	563316	566711	570116	573531	576956	580391	583836	587291	590756	594231	597716	601211	604716	608231	611756	615291	618836	622391	625956	629531	633116	636711	640316	643931	647556	651191	654836	658491	662156	665831	669516	673211	676916	680631	684356	688091	691836	695591	699356	703131	706916	710711	714516	718331	722156	725991	729836	733691	737556	741431	745316	749211	753116	757031	760956	764891	768836	772791	776756	780731	784716	788711	792716	796731	800756	804791	808836	812891	816956	821031	825116	829211	833316	837431	841556	845691	849836	853991	858156	862331	866516	870711	874916	879131	883356	887591	891836	896091	900356	904631	908916	913211	917516	921831	926156	930491	934836	939191	943556	947931	952316	956711	961116	965531	969956	974391	978836	983291	987756	992231	996716	1001211	1005716	1010231	1014756	1019291	1023836	1028391	1032956	1037531	1042116	1046711	1051316	1055931	1060556	1065191	1069836	1074491	1079156	1083831	1088516	1093211	1097916	1102631	1107356	1112091	1116836	1121591	1126356	1131131	1135916	1140711	1145516	1150331	1155156	1159991	1164836	1169691	1174556	1179431	1184316	1189211	1194116	1199031	1203956	1208891	1213836	1218791	1223756	1228731	1233716	1238711	1243716	1248731	1253756	1258791	1263836	1268891	1273956	1279031	1284116	1289211	1294316	1299431	1304556	1309691	1314836	1319991	1325156	1330331	1335516	1340711	1345916	1351131	1356356	1361591	1366836	1372091	1377356	1382631	1387916	1393211	1398516	1403831	1409156	1414491	1419836	1425191	1430556	1435931	1441316	1446711	1452116	1457531	1462956	1468391	1473836	1479291	1484756	1490231	1495716	1501211	1506716	1512231	1517756	1523291	1528836	1534391	1539956	1545531	1551116	1556711	1562316	1567931	1573556	1579191	1584836	1590491	1596156	1601831	1607516	1613211	1618916	1624631	1630356	1636091	1641836	1647591	1653356	1659131	1664916	1670711	1676516	1682331	1688156	1694091	1700036	1705991	1711956	1717931	1723916	1729911	1735916	1741931	1747956	1753991	1759036	1765091	1771156	1777231	1783316	1789411	1795516	1801631	1807756	1813891	1819036	1825191	1831356	1837531	1843716	1849911	1856116	1862331	1868556	1874791	1881036	1887291	1893556	1899831	1906116	1912411	1918716	1925031	1931356	1937691	1944036	1950391	1956756	1963131	1969516	1975911	1982316	1988731	1995156	2001591	2008036	2014491	2020956	2027431	2033916	2040411	2046916	2053431	2059956	2066491	2073036	2079591	2086156	2092731	2099316	2105911	2112516	2119131	2125756	2132391	2139036	2145691	2152356	2159031	2165716	2172411	2179116	2185831	2192556	2199291	2206036	2212791	2219556	2226331	2233116	2239911	2246716	2253531	2260356	2267191	2274036	2280891	2287756	2294631	2301516	2308411	2315316	2322231	2329156	2336091	2343036	2349991	2356956	2363931	2370916	2377911	2384916	2391931	2398956	2405991	2413036	2420091	2427156	2434231	2441316	2448411	2455516	2462631	2469756	2476891	2484036	2491191	2498356	2505531	2512716	2519911	2527116	2534331	2541556	2548791	2556036	2563291	2570556	2577831	2585116	2592411	2600716	2608031	2615356	2622691	2629036	2636391	2643756	2651131	2658516	2665911	2673316	2680731	2688156	2695591	2703036	2710491	2717956	2725431	2732916	2740411	2747916	2755431	2762956	2770491	2778036	2785591	2793156	2800731	2808316	2815911	2823516	2831131	2838756	2846391	2854036	2861691	2869356	2877031	2884716	2892411	2900116	2907831	2915556	2923291	2931036	2938791	2946556	2954331	2962116	2969911	2977716	2985531	2993356	3001191	3009036	3016891	3024756	3032631	3040516	3048411	3056316	3064231	3072156	3080091	3088036	3095991	3103956	3111931	3119916	3127911	3135916	3143931	3151956	3159991	3168036	3176091	3184156	3192231	3200316	3208411	3216516	3224631	3232756	3240891	3249036	3257191	3265356	3273531	3281716	3289911	3298116	3306331	3314556	3322791	3331036	3339291	3347556	3355831	3364116	3372411	3380716	3389031	3397356	3405691	3414036	3422391	3430756	3439131	3447516	3455911	3464316	3472731	3481156	3489591	3498036	3506491	3514956	3523431	3531916	3540411	3548916	3557431	3565956	3574491	3583036	3591591	3600156	3608731	3617316	3625911	3634516	3643131	3651756	3660391	3669036	3677691	3686356	3695031	3703716	3712411	3721116	3729831	3738556	3747291	3756036	3764791	3773556	3782331	3791116	3800911	3809716	3818531	3827356	3836191	3845036	3853891	3862756	3871631	3880516	3889411	3898316	3907231	3916156	3925091	3934036	3943031	3952036	3961091	3970156	3979231	3988316	3997411	4006516	4015631	4024756	4033891	4043036	4052191	4061356	4070531	4079716	4088911	4098116	4107331	4116556	4125791	4135036	4144291	4153556	4162831	4172116	4181411	4190716	4200031	4209356	4218691	4228036	4237391	4246756	4256131	4265516	4274911	4284316	4293731	4303156	4312591	4322036	4331491	4340956	4350431	4359916	4369411	4378916	4388431	4397956	4407491	4417036	4426591	4436156	4445731	4455316	4464911	4474516	4484131	4493756	4503391	4513036	4522691	4532356	4542031	4551716	4561411	4571116	4580831	4590556	4600291	4610036	4619791	4629556	4639331	4649116	4658911	4668716	4678531	4688356	4698191	4708036	4717891	4727756	4737631	4747516

H06

MACY:DEC-11-DCFPAB-B TEST OF DCFPS, STFPS, SETF, SETD, SETI, SETL, CFCC
DCFPAB.F11 CROSS REFERENCE TABLE -- PERMANENT SYMBOLS

MACY1: 27'732, 10-SEP-76 11:04 PAGE 75

DEFAULT GLOBALS GENERATED: 0

*DCFPAB DCFPAB, SEQ 50L CRF DS:ERFZ EN:ABS=DSKM:DCFPAB.F11
MACY: 11:04 4 SECONDS
CPU TIME: 12 SECS
JOB: 124 PAGES

