FP11

LDCJX, STCXJ

MD-11-DCFPJ-B

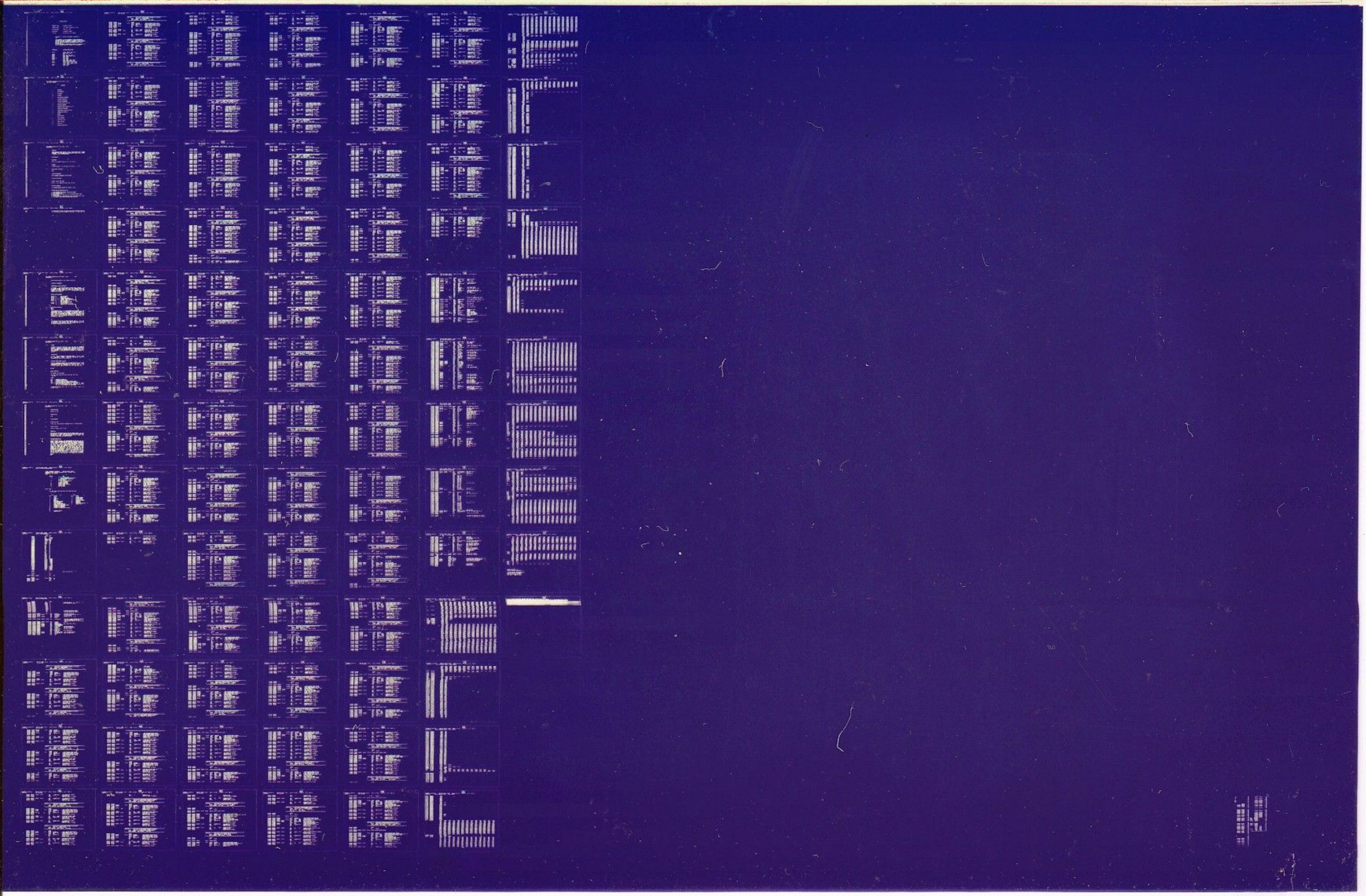
EP-DCFPJ-B-DL-A

OCT 1976

digital

FICHE 1 OF 1

Made in U.S.A.



.REPT 0

IDENTIFICATION

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

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

TABLE OF CONTENTS

CONTENTS

FP11 BASIC INSTRUCTION TEST DCFPA - DCFPL PAGE 3
DESCRIPTION

1. ABSTRACT

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

2. REQUIREMENTS

2.1 EQUIPMENT

PDP11/45 STANDARD COMPUTER WITH FP11 OPTION

2.2 STORAGE

PROGRAM STORAGE - THE ROUTINES USE MEMORY 0 - 17776

2.3 PRELIMINARY PROGRAMS

NONE

3. LOADING PROCEDURE

USE STANDARD PROCEDURE FOR ABS TAPES.

4. STARTING PROCEDURE

4.1 CONTROL SWITCH SETTINGS

SEE 5.1.1 (ALL DOWN FOR WORST CASE TESTING.

4.2 STARTING ADDRESS

THE PROGRAM SHOULD ALWAYS BE STARTED AT 200.

4.3 PROGRAM AND/OR OPERATOR ACTION

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

E01

[illegible]

[illegible] THE DISPLAY ON THE 11/45 WILL SHOW THE ITERATION [illegible]
[illegible] THE LEFT BYTE AND TEST NUMBER IN THE RIGHT. [illegible] SEE. SEE THE

DATA DISPLAY SWITCH TO THE DISPLAY POSITION.

5. OPERATING PROCEDURE

5.1 OPERATIONAL SWITCH SETTINGS

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

5.1.1 SWITCH SETTINGS ARE:

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

5.2 SUBROUTINE ABSTRACTS

5.2.1 SCOPE

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

5.2.2 HLT

THIS ROUTINE PRINTS OUT AN ERROR MESSAGE (SEE 6.1.) IF A HLT IS EXECUTED. THE SUBTEST WILL BE LOOPED UPON UNTIL 256 CONSECUTIVE GOOD PASSES ARE COMPLETED IF SW<9> IS ON A 1. TO INHIBIT TYPEOUTS, PUT SW<12> ON A 1.

5.2.3 TRTRAP

IF SW(12) IS ON A 0, THE T BIT WILL BE SET ON ALTERNATE PASSES. WHEN SET, IT CAUSES A TRAP AFTER EACH INSTRUCTION. THE FIRST INSTRUCTION EXECUTED UPON TRAPPING IS AN "RTT" WHICH RETURNS TO THE INTERRUPTED SEQUENCE OF INSTRUCTIONS. THIS SEQUENCE IS CONTINUED UNTIL THE END OF THE PROGRAM IS REACHED.

5.2.4 TRAPCATCHER

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

5.2.5 FLOATING POINT TRAP (TO 244)

THE FP11 INTERRUPT DISABLE BIT IS ALWAYS SET IN ALL OF THESE TESTS (EXCEPT DCFPA) SO NO TRAPS TO 244 SHOULD OCCUR. IF AN INTERRUPT OCCURS, THE PROGRAM WILL HALT AT 766 IN THE ROUTINE CALLED FLTERR AND DISPLAY THE FPS REGISTER IN R0.

6. ERRORS

6.1 ERROR PRINTOUT

THE FORMAT IS AS FOLLOWS:

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

WHERE:

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

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

6.2 ERROR RECOVERY

RESTART AT 200

7. RESTRICTIONS

NONE

8. MISCELLANEOUS

8.1 EXECUTION TIME

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

8.2 STACK POINTER

STACK IS INITALLY SET TO 600

8.3 POWER FAIL

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

9. PROGRAM DESCRIPTION

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

.END

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

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

OUTPUT FORM:

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

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

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

                        000000  .=      0                       ;TRAP CATCHER FROM 0 - 776

                        000200  .=      200

000200  000167  000622                  JMP     BEG

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

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

001026  012706  000600        BEG:    MOV     #600,SP         ;** STACK AT 500 **
001032  012737  001054  000004        MOV     #M1120,@#4      ;FIND OUT WHICH MACHINE THIS IS
001040  005737  177772              TST     @#177772        ;IS PIRQ THERE?
001044  012767  000006  015206      MOV     #6,YESRT        ;FUDGE IN RTT IF 11/45
001052  000403                      BR      BEGIN

001054  016737  016342  000C10  M1120:  MOV     FPTADR,@#10     ;LOAD THE ILLEGAL INSTRUCTION VECTOR
                                                              ; WITH THE ADDRESS OF THE FPU.
                                                              ; THE FPU WILL HANDLE THE BAD OPCODES
001062  012737  000006  000004  BEGIN:  MOV     #6,@#4          ;RESET 4
001070  012706  000600              MOV     #600,SP
001074  012737  016260  000014      MOV     #YESRT,@#14     ;SET TRACE TRAP VECTOR
001102  012777  017120  016320      MOV     #POWDWN,@DWNVEC
001110  012777  000340  016314      MOV     #340,@DWNVEC+2
001116  012737  017320  000020      MOV     #.IOT,@#20      ;SET UP VECTOR 20
001124  012700  000030              MOV     #30,R0          ;SET R0 TO VECTOR 30
001130  012720  016422              MOV     #.TRP,(0)+      ;SET EMT VECTOR
001134  012720  000340              MOV     #340,(0)+
001140  012720  016262              MOV     #.EMT,(0)+      ;SET TRAP VECTOR
001144  012710  000340              MOV     #340,(0)
001150  012777  000760  016246      MOV     #FLTERR,@FPVECT ;LOAD INTERUPT VECTOR
001156  012777  000340  016242      MOV     #340,@FPVECT+2  ;LOCK UP PROCESSOR
001164  005067  177610              CLR     ICNT
001170  005067  016250              CLR     LAD
```

```
                                ;*****************************************************************
                                ;TEST 1:        TEST LDCIF (LOAD-CONVERT INTEGER TO FLOATING)
                                ;       LOAD    000000 --> 000000,000000
                                ;       FPS = 047404,   SRC = M2-R7,    AC = AC1
                                ;*****************************************************************

001174  104400                          SCOPE
001176  170127  047400          TST1:   LDFPS   #047400         ;LOAD FLOATING POINT STATUS
001202  177127  000000                  LDCIF   #000000,AC1     ;LOAD-CONVERT 000000 INTO AC1
001206  170200                          STFPS   FPS             ;STORE FLOATING POINT STATUS
001210  022700  047404                  CMP     #047404,FPS     ;CHECK FLOATING POINT STATUS
001214  001401                          BEQ     .+4             ;BRANCH IF OK
001216  104000                          HLT                     ;FPS NOT EQUAL TO 047404

001220  174167  177556                  STF     AC1,    ANS1    ;STORE AC1 IN ANS1, ANS2
001224  022767  000000  177550          CMP     #000000,ANS1    ;DID 000000 GET STORED?
001232  001401                          BEQ     .+4             ;BRANCH IF OK
001234  104002                          HLT+2                   ;ANS1 NOT EQUAL TO 000000

001236  022767  000000  177540          CMP     #000000,ANS2    ;DID 000000 GET STORED?
001244  001401                          BEQ     .+4             ;BRANCH IF OK
001246  104002                          HLT+2                   ;ANS2 NOT EQUAL TO 000000


                                ;*****************************************************************
                                ;TEST 2:        TEST LDCIF (LOAD-CONVERT INTEGER TO FLOATING)
                                ;       LOAD    125252 --> 143652,126000
                                ;       FPS = 047400,   SRC = M2-R7,    AC = AC1
                                ;*****************************************************************

001250  104400                          SCOPE
001252  170127  047400          TST2:   LDFPS   #047400         ;LOAD FLOATING POINT STATUS
001256  177127  125252                  LDCIF   #125252,AC1     ;LOAD-CONVERT 125252 INTO AC1
001262  170200                          STFPS   FPS             ;STORE FLOATING POINT STATUS
001264  022700  047410                  CMP     #047410,FPS     ;CHECK FLOATING POINT STATUS
001270  001401                          BEQ     .+4             ;BRANCH IF OK
001272  104000                          HLT                     ;FPS NOT EQUAL TO 047410

001274  174167  177502                  STF     AC1,    ANS1    ;STORE AC1 IN ANS1, ANS2
001300  022767  143652  177474          CMP     #143652,ANS1    ;DID 143652 GET STORED?
001306  001401                          BEQ     .+4             ;BRANCH IF OK
001310  104002                          HLT+2                   ;ANS1 NOT EQUAL TO 143652

001312  022767  126000  177464          CMP     #126000,ANS2    ;DID 126000 GET STORED?
001320  001401                          BEQ     .+4             ;BRANCH IF OK
001322  104002                          HLT+2                   ;ANS2 NOT EQUAL TO 126000


                                ;*****************************************************************
                                ;TEST 3:        TEST LDCIF (LOAD-CONVERT INTEGER TO FLOATING)
                                ;       LOAD    052525 --> 043652,125000
                                ;       FPS = 047400,   SRC = M2-R7,    AC = AC0
                                ;*****************************************************************

001324  104400                          SCOPE
```

```
001326  170127  047400          TST3:   LDFPS   #047400          ;LOAD FLOATING POINT STATUS
001332  177027  052525                  LDCIF   #052525,AC0       ;LOAD-CONVEFT 052525 INTO AC0
001336  170200                          STFPS   FPS               ;STORE FLOATING POINT STATUS
001340  022700  047400                  CMP     #047400,FPS       ;CHECK FLOATING POINT STATUS
001344  001401                          BEQ     .+4               ;BRANCH IF OK
001346  104000                          HLT                       ;FPS NOT EQUAL TO 047400

001350  174067  177426                  STF     AC0,    ANS1      ;STORE AC0 IN ANS1, ANS2
001354  022767  043652  177420          CMP     #043652,ANS1      ;DID 043652 GET STORED?
001362  001401                          BEQ     .+4               ;BRANCH IF OK
001364  104002                          HLT+2                     ;ANS1 NOT EQUAL TO 043652

001366  022767  125000  177410          CMP     #125000,ANS2      ;DID 125000 GET STORED?
001374  001401                          BEQ     .+4               ;BRANCH IF OK
001376  104002                          HLT+2                     ;ANS2 NOT EQUAL TO 125000


                        ;*****************************************************************
                        ;TEST 4:        TEST LDCIF (LOAD-CONVERT INTEGER TO FLOATING)
                        ;       LOAD    000001 --> 040200,000000
                        ;       FPS = 047400,   SRC = M2-R7,    AC = AC0
                        ;*****************************************************************

001400  104400                          SCOPE
001402  170127  047400          TST4:   LDFPS   #047400           ;LOAD FLOATING POINT STATUS
001406  177027  000001                  LDCIF   #000001,AC0       ;LOAD-CONVERT 000001 INTO AC0
001412  170200                          STFPS   FPS               ;STORE FLOATING POINT STATUS
001414  022700  047400                  CMP     #047400,FPS       ;CHECK FLOATING POINT STATUS
001420  001401                          BEQ     .+4               ;BRANCH IF OK
001422  104000                          HLT                       ;FPS NOT EQUAL TO 047400

001424  174067  177352                  STF     AC0,    ANS1      ;STORE AC0 IN ANS1, ANS2
001430  022767  040200  177344          CMP     #040200,ANS1      ;DID 040200 GET STORED?
001436  001401                          BEQ     .+4               ;BRANCH IF OK
001440  104002                          HLT+2                     ;ANS1 NOT EQUAL TO 040200

001442  022767  000000  177334          CMP     #000000,ANS2      ;DID 000000 GET STORED?
001450  001401                          BEQ     .+4               ;BRANCH IF OK
001452  104002                          HLT+2                     ;ANS2 NOT EQUAL TO 000000


                        ;*****************************************************************
                        ;TEST 5:        TEST LDCIF (LOAD-CONVERT INTEGER TO FLOATING)
                        ;       LOAD    177777 --> 140200,000000
                        ;       FPS = 047410,   SRC = M2-R7,    AC = AC3
                        ;*****************************************************************

001454  104400                          SCOPE
001456  170127  047400          TST5:   LDFPS   #047400           ;LOAD FLOATING POINT STATUS
001462  177327  177777                  LDCIF   #177777,AC3       ;LOAD-CONVERT 177777 INTO AC3
001466  170200                          STFPS   FPS               ;STORE FLOATING POINT STATUS
001470  022700  047410                  CMP     #047410,FPS       ;CHECK FLOATING POINT STATUS
001474  001401                          BEQ     .+4               ;BRANCH IF OK
001476  104000                          HLT                       ;FPS NOT EQUAL TO 047410

001500  174367  177276                  STF     AC3,    ANS1      ;STORE AC3 IN ANS1, ANS2
```

```
001504  022767  140200  177270          CMP     #140200,ANS1    ;DID 140200 GET STORED?
001512  001401                          BEQ     .+4             ;BRANCH IF OK
001514  104002                          HLT+2                   ;ANS1 NOT EQUAL TO 140200

001516  022767  000000  177260          CMP     #000000,ANS2    ;DID 000000 GET STORED?
001524  001401                          BEQ     .+4             ;BRANCH IF OK
001526  104002                          HLT+2                   ;ANS2 NOT EQUAL TO 000000


                        ;*****************************************************************
                        ;TEST 6:        TEST LDCIF (LOAD-CONVERT INTEGER TO FLOATING)
                        ;       LOAD    100000 --> 144000,000000
                        ;       FPS = 047410,   SRC = M2-R7,    AC = AC0
                        ;*****************************************************************

001530  104400                          SCOPE
001532  170127  047400          TST6:   LDFPS   #047400         ;LOAD FLOATING POINT STATUS
001536  177027  100000                  LDCIF   #100000,AC0     ;LOAD-CONVERT 100000 INTO AC0
001542  170200                          STFPS   FPS             ;STORE FLOATING POINT STATUS
001544  022700  047410                  CMP     #047410,FPS     ;CHECK FLOATING POINT STATUS
001550  001401                          BEQ     .+4             ;BRANCH IF OK
001552  104000                          HLT                     ;FPS NOT EQUAL TO 047410

001554  174067  177222                  STF     AC0,    ANS1    ;STORE AC0 IN ANS1, ANS2
001560  022767  144000  177214          CMP     #144000,ANS1    ;DID 144000 GET STORED?
001566  001401                          BEQ     .+4             ;BRANCH IF OK
001570  104002                          HLT+2                   ;ANS1 NOT EQUAL TO 144000

001572  022767  000000  177204          CMP     #000000,ANS2    ;DID 000000 GET STORED?
001600  001401                          BEQ     .+4             ;BRANCH IF OK
001602  104002                          HLT+2                   ;ANS2 NOT EQUAL TO 000000


                        ;*****************************************************************
                        ;TEST 7:        TEST LDCIF (LOAD-CONVERT INTEGER TO FLOATING)
                        ;       LOAD    077777 --> 043777,177000
                        ;       FPS = 047400,   SRC = M2-R7,    AC = AC2
                        ;*****************************************************************

001604  104400                          SCOPE
001606  170127  047400          TST7:   LDFPS   #047400         ;LOAD FLOATING POINT STATUS
001612  177227  077777                  LDCIF   #077777,AC2     ;LOAD-CONVERT 077777 INTO AC2
001616  170200                          STFPS   FPS             ;STORE FLOATING POINT STATUS
001620  022700  047400                  CMP     #047400,FPS     ;CHECK FLOATING POINT STATUS
001624  001401                          BEQ     .+4             ;BRANCH IF OK
001626  104000                          HLT                     ;FPS NOT EQUAL TO 047400

001630  174267  177146                  STF     AC2,    ANS1    ;STORE AC2 IN ANS1, ANS2
001634  022767  043777  177140          CMP     #043777,ANS1    ;DID 043777 GET STORED?
001642  001401                          BEQ     .+4             ;BRANCH IF OK
001644  104002                          HLT+2                   ;ANS1 NOT EQUAL TO 043777

001646  022767  177000  177130          CMP     #177000,ANS2    ;DID 177000 GET STORED?
001654  001401                          BEQ     .+4             ;BRANCH IF OK
001656  104002                          HLT+2                   ;ANS2 NOT EQUAL TO 177000
```

```
                        ;*****************************************************************
                        ;TEST 10:       TEST LDCIF (LOAD-CONVERT INTEGER TO FLOATING)
                        ;       LOAD    000125 --> 041652,000000
                        ;       FPS = 047400,   SRC = M2-R7,    AC = AC2
                        ;*****************************************************************

001660  104400                  SCOPE
001662  170127  047400  TST10:  LDFPS   #047400          ;LOAD FLOATING POINT STATUS
001666  177227  000125          LDCIF   #000125,AC2     ;LOAD-CONVERT 000125 INTO AC2
001672  170200                  STFPS   FPS             ;STORE FLOATING POINT STATUS
001674  022700  047400          CMP     #047400,FPS     ;CHECK FLOATING POINT STATUS
001700  001401                  BEQ     .+4             ;BRANCH IF OK
001702  104000                  HLT                     ;FPS NOT EQUAL TO 047400

001704  174267  177072          STF     AC2,    ANS1    ;STORE AC2 IN ANS1, ANS2
001710  022767  041652  177064  CMP     #041652,ANS1    ;DID 041652 GET STORED?
001716  001401                  BEQ     .+4             ;BRANCH IF OK
001720  104002                  HLT+2                   ;ANS1 NOT EQUAL TO 041652

001722  022767  000000  177054  CMP     #000000,ANS2    ;DID 000000 GET STORED?
001730  001401                  BEQ     .+4             ;BRANCH IF OK
001732  104002                  HLT+2                   ;ANS2 NOT EQUAL TO 000000

                        ;*****************************************************************
                        ;TEST 11:       TEST LDCIF (LOAD-CONVERT INTEGER TO FLOATING)
                        ;       LOAD    070707 --> 043743,107000
                        ;       FPS = 047400,   SRC = M0-R2,    AC = AC1
                        ;*****************************************************************

001734  104400                  SCOPE
001736  170127  047400  TST11:  LDFPS   #047400          ;LOAD FLOATING POINT STATUS
001742  012702  070707          MOV     #070707,R2      ;LOAD 070707 INTO R2
001746  177102                  LDCIF   R2,     AC1     ;LOAD-CONVERT 070707 INTO AC1
001750  170200                  STFPS   FPS             ;STORE FLOATING POINT STATUS
001752  022700  047400          CMP     #047400,FPS     ;CHECK FLOATING POINT STATUS
001756  001401                  BEQ     .+4             ;BRANCH IF OK
001760  104000                  HLT                     ;FPS NOT EQUAL TO 047400

001762  174167  177014          STF     AC1,    ANS1    ;STORE AC1 IN ANS1, ANS2
001766  022767  043743  177006  CMP     #043743,ANS1    ;DID 043743 GET STORED?
001774  001401                  BEQ     .+4             ;BRANCH IF OK
001776  104002                  HLT+2                   ;ANS1 NOT EQUAL TO 043743

002000  022767  107000  176776  CMP     #107000,ANS2    ;DID 107000 GET STORED?
002006  001401                  BEQ     .+4             ;BRANCH IF OK
002010  104002                  HLT+2                   ;ANS2 NOT EQUAL TO 107000

                        ;*****************************************************************
                        ;TEST 12:       TEST LDCIF (LOAD-CONVERT INTEGER TO FLOATING)
                        ;       LOAD    107070 --> 143743,110000
                        ;       FPS = 047410,   SRC = M6-R7,    AC = AC2
                        ;*****************************************************************
```

```
002012  104400                          SCOPE
002014  000401                          BR      TST12           ;SKIP DATA

002016  107070          DAT12:  107070

002020  170127  047400  TST12:  LDFPS   #047400         ;LOAD FLOATING POINT STATUS
002024  177267  177766          LDCIF   DAT12,  AC2     ;LOAD-CONVERT 107070 INTO AC2
002030  170200                  STFPS   FPS             ;STORE FLOATING POINT STATUS
002032  022700  047410          CMP     #047410,FPS     ;CHECK FLOATING POINT STATUS
002036  001401                  BEQ     .+4             ;BRANCH IF OK
002040  104000                  HLT                     ;FPS NOT EQUAL TO 047410

002042  174267  176734          STF     AC2,    ANS1    ;STORE AC2 IN ANS1, ANS2
002046  022767  143743  176726  CMP     #143743,ANS1    ;DID 143743 GET STORED?
002054  001401                  BEQ     .+4             ;BRANCH IF OK
002056  104002                  HLT+2                   ;ANS1 NOT EQUAL TO 143743

002060  022767  110000  176716  CMP     #110000,ANS2    ;DID 110000 GET STORED?
002066  001401                  BEQ     .+4             ;BRANCH IF OK
002070  104002                  HLT+2                   ;ANS2 NOT EQUAL TO 110000


                        ;*****************************************************************
                        ;TEST 13:       TEST STCFI (STORE-CONVERT FLOATING TO INTEGER.
                        ;       STORE   000000.000000 --> 000000
                        ;       FPS = 047404,   AC = AC0,       DST = M6-R7
                        ;*****************************************************************

002072  104400                          SCOPE
002074  000402                          BR      TST13

002076  000000  000000  DAT13:  000000.000000

002102  170127  047400  TST13:  LDFPS   #047400         ;LOAD FLOATING POINT STATUS
002106  172467  177764          LDF     DAT13,  AC0     ;LOAD 000000.000000 INTO AC0
002112  175467  176664  FPI13:  STCFI   AC0,    ANS1    ;STORE-CONVERT AC0 IN ANS1
002116  013767  177776  176660  MOV     @#PS,   ANS2    ;GET CPU STATUS
002124  042767  177760  176652  BIC     #177760,ANS2    ;SAVE CONDITION CODES
002132  170200                  STFPS   FPS             ;STORE FLOATING POINT STATUS
002134  022700  047404          CMP     #047404,FPS     ;CHECK FLOATING POINT STATUS
002140  001401                  BEQ     .+4             ;BRANCH IF OK
002142  104000                  HLT                     ;FPS NOT EQUAL TO 047404

002144  022767  000000  176630  CMP     #000000,ANS1    ;DID 000000 GET STORED?
002152  001401                  BEQ     .+4             ;BRANCH IF OK
002154  104001                  HLT+1                   ;ANS1 NOT EQUAL TO 000000

002156  022767  000004  176620  CMP     #4,     ANS2    ;CONDITION CODES = 4?
002164  001401                  BEQ     .+4             ;BRANCH IF OK
002166  104002                  HLT+2                   ;WRONG CONDITION CODES!


                        ;*****************************************************************
                        ;TEST 14:       TEST STCFI (STORE-CONVERT FLOATING TO INTEGER
                        ;       STORE   041252.125252 --> 000025
                        ;       FPS = 047400,   AC = AC0,       DST = M6-R7
```

```
                        ;*****************************************************************************
002170  104400                    SCOPE
002172  000402                    BR      TST14

002174  041252  125252    DAT14:  041252,125252

002200  170127  047400    TST14:  LDFPS   #047400         ;LOAD FLOATING POINT STATUS
002204  172467  177764            LDF     DAT14,  AC0     ;LOAD 041252,125252 INTO AC0
002210  175467  176566    FPI14:  STCFI   AC0,    ANS1    ;STORE-CONVERT AC0 IN ANS1
002214  013767  177776  176562    MOV     @#PS,   ANS2    ;GET CPU STATUS
002222  042767  177760  176554    BIC     #177760,ANS2    ;SAVE CONDITION CODES
002230  170200                    STFPS   FPS             ;STORE FLOATING POINT STATUS
002232  022700  047400            CMP     #047400,FPS     ;CHECK FLOATING POINT STATUS
002236  001401                    BEQ     .+4             ;BRANCH IF OK
002240  104000                    HLT                     ;FPS NOT EQUAL TO 047400

002242  022767  000025  176532    CMP     #000025,ANS1    ;DID 000025 GET STORED?
002250  001401                    BEQ     .+4             ;BRANCH IF OK
002252  104001                    HLT+1                   ;ANS1 NOT EQUAL TO 000025

002254  022767  000000  176522    CMP     #0,     ANS2    ;CONDITION CODES = 0?
002262  001401                    BEQ     .+4             ;BRANCH IF OK
002264  104002                    HLT+2                   ;WRONG CONDITION CODES!

                        ;*****************************************************************************
                        ;TEST 15:      TEST STCFI (STORE-CONVERT FLOATING TO INTEGER)
                        ;       STORE   141252,125252 --> 177753
                        ;       FPS = 047410,   AC = AC3,       DST = M6-R7
                        ;*****************************************************************************

002266  104400                    SCOPE
002270  000402                    BR      TST15

002272  141252  125252    DAT15:  141252,125252

002276  170127  047400    TST15:  LDFPS   #047400         ;LOAD FLOATING POINT STATUS
002302  172767  177764            LDF     DAT15,  AC3     ;LOAD 141252,125252 INTO AC3
002306  175767  176470    FPI15:  STCFI   AC3,    ANS1    ;STORE-CONVERT AC3 IN ANS1
002312  013767  177776  176464    MOV     @#PS,   ANS2    ;GET CPU STATUS
002320  042767  177760  176456    BIC     #177760,ANS2    ;SAVE CONDITION CODES
002326  170200                    STFPS   FPS             ;STORE FLOATING POINT STATUS
002330  022700  047410            CMP     #047410,FPS     ;CHECK FLOATING POINT STATUS
002334  001401                    BEQ     .+4             ;BRANCH IF OK
002336  104000                    HLT                     ;FPS NOT EQUAL TO 047410

002340  022767  177753  176434    CMP     #177753,ANS1    ;DID 177753 GET STORED?
002346  001401                    BEQ     .+4             ;BRANCH IF OK
002350  104001                    HLT+1                   ;ANS1 NOT EQUAL TO 177753

002352  022767  000010  176424    CMP     #10.    ANS2    ;CONDITION CODES = 10?
002360  001401                    BEQ     .+4             ;BRANCH IF OK
002362  104002                    HLT+2                   ;WRONG CONDITION CODES!
```

```
                        ;*****************************************************************
                        ;TEST 16:       TEST STCFI (STORE-CONVERT FLOATING TO INTEGER)
                        ;       STORE   040177,177777 --> 000000
                        ;       FPS = 047404,   AC = AC0,       DST = M6-R7
                        ;*****************************************************************

002364  104433                   SCOPE
002366  000402                   BR      TST16

002370  040177  177777   DAT16:  040177,177777

002374  170127  047400   TST16:  LDFPS   #047400         ;LOAD FLOATING POINT STATUS
002400  172467  177764           LDF     DAT16,  AC0     ;LOAD 040177,177777 INTO AC0
002404  175467  176372   FPI16:  STCFI   AC0,    ANS1    ;STORE-CONVERT AC0 IN ANS1
002410  013767  177776  176366   MOV     @#PS,   ANS2    ;GET CPU STATUS
002416  042767  177760  176360   BIC     #177760,ANS2    ;SAVE CONDITION CODES
002424  170200                   STFPS   FPS             ;STORE FLOATING POINT STATUS
002426  022700  047404           CMP     #047404,FPS     ;CHECK FLOATING POINT STATUS
002432  001401                   BEQ     .+4             ;BRANCH IF OK
002434  104000                   HLT                     ;FPS NOT EQUAL TO 047404

002436  022767  000000  176336   CMP     #000000,ANS1    ;DID 000000 GET STORED?
002444  001401                   BEQ     .+4             ;BRANCH IF OK
002446  104001                   HLT+1                   ;ANS1 NOT EQUAL TO 000000

002450  022767  000004  176326   CMP     #4,     ANS2    ;CONDITION CODES = 4?
002456  001401                   BEQ     .+4             ;BRANCH IF OK
002460  104002                   HLT+2                   ;WRONG CONDITION CODES!


                        ;*****************************************************************
                        ;TEST 17:       TEST STCFI (STORE-CONVERT FLOATING TO INTEGER
                        ;       STORE   140177,177777 --> 000000
                        ;       FPS = 047404,   AC = AC2,       DST = M6-R7
                        ;*****************************************************************

002462  104433                   SCOPE
002464  000402                   BR      TST17

002466  140177  177777   DAT17:  140177,177777

002472  170127  047400   TST17:  LDFPS   #047400         ;LOAD FLOATING POINT STATUS
002476  172667  177764           LDF     DAT17,  AC2     ;LOAD 140177,177777 INTO AC2
002502  175667  176274   FPI17:  STCFI   AC2,    ANS1    ;STORE-CONVERT AC2 IN ANS1
002506  013767  177776  176270   MOV     @#PS,   ANS2    ;GET CPU STATUS
002514  042767  177760  176262   BIC     #177760,ANS2    ;SAVE CONDITION CODES
002522  170200                   STFPS   FPS             ;STORE FLOATING POINT STATUS
002524  022700  047404           CMP     #047404,FPS     ;CHECK FLOATING POINT STATUS
002530  001401                   BEQ     .+4             ;BRANCH IF OK
002532  104000                   HLT                     ;FPS NOT EQUAL TO 047404

002534  022767  000000  176240   CMP     #000000,ANS1    ;DID 000000 GET STORED?
002542  001401                   BEQ     .+4             ;BRANCH IF OK
002544  104001                   HLT+1                   ;ANS1 NOT EQUAL TO 000000

002546  022767  000004  176232   CMP     #4,     ANS2    ;CONDITION CODES = 4?
```

```
002554  001401                          BEQ     .+4             ;BRANCH IF OK
002556  104002                          HLT+2                   ;WRONG CONDITION CODES!


                        ;*****************************************************************
                        ;TEST 20:       TEST STCFI (STORE-CONVERT FLOATING TO INTEGER)
                        ;       STORE   040200,000001 --> 000001
                        ;       FPS = 047400,   AC = AC2,       DST = M6-R7
                        ;*****************************************************************

002560  104400                          SCOPE
002562  000402                          BR      TST20

002564  040200  000001        DAT20:  040200,000001

002570  170127  047400        TST20:  LDFPS   #047400         ;LOAD FLOATING POINT STATUS
002574  172667  177764                LDF     DAT20,  AC2     ;LOAD 040200,000001 INTO AC2
002600  175667  176176        FPI20:  STCFI   AC2,    ANS1    ;STORE-CONVERT AC2 IN ANS1
002604  013767  177776  176172        MOV     @#PS,   ANS2    ;GET CPU STATUS
002612  042767  177760  176164        BIC     #177760,ANS2    ;SAVE CONDITION CODES
002620  170200                        STFPS   FPS             ;STORE FLOATING POINT STATUS
002622  022700  047400                CMP     #047400,FPS     ;CHECK FLOATING POINT STATUS
002626  001401                        BEQ     .+4             ;BRANCH IF OK
002630  104000                        HLT                     ;FPS NOT EQUAL TO 047400

002632  022767  000001  176142        CMP     #000001,ANS1    ;DID 000001 GET STORED?
002640  001401                        BEQ     .+4             ;BRANCH IF OK
002642  104001                        HLT+1                   ;ANS1 NOT EQUAL TO 000001

002644  022767  000000  176132        CMP     #0,     ANS2    ;CONDITION CODES = 0?
002652  001401                        BEQ     .+4             ;BRANCH IF OK
002654  104002                        HLT+2                   ;WRONG CONDITION CODES!


                        ;*****************************************************************
                        ;TEST 21:       TEST STCFI (STORE-CONVERT FLOATING TO INTEGER)
                        ;       STORE   140200,000001 --> 177777
                        ;       FPS = 047410,   AC = AC1,       DST = M6-R7
                        ;*****************************************************************

002656  104400                          SCOPE
002660  000402                          BR      TST21

002662  140200  000001        DAT21:  140200,000001

002666  170127  047400        TST21:  LDFPS   #047400         ;LOAD FLOATING POINT STATUS
002672  172567  177764                LDF     DAT21,  AC1     ;LOAD 140200,000001 INTO AC1
002676  175567  176100        FPI21:  STCFI   AC1,    ANS1    ;STORE-CONVERT AC1 IN ANS1
002702  013767  177776  176074        MOV     @#PS,   ANS2    ;GET CPU STATUS
002710  042767  177760  176066        BIC     #177760,ANS2    ;SAVE CONDITION CODES
002716  170200                        STFPS   FPS             ;STORE FLOATING POINT STATUS
002720  022700  047410                CMP     #047410,FPS     ;CHECK FLOATING POINT STATUS
002724  001401                        BEQ     .+4             ;BRANCH IF OK
002726  104000                        HLT                     ;FPS NOT EQUAL TO 047410

002730  022767  177777  176044        CMP     #177777,ANS1    ;DID 177777 GET STORED?
```

```
002736  001401                          BEQ     .+4             ;BRANCH IF OK
002740  104001                          HLT+1                   ;ANS1 NOT EQUAL TO 177777

002742  022767  000010  176034          CMP     #10.    ANS2    ;CONDITION CODES = 10?
002750  001401                          BEQ     .+4             ;BRANCH IF OK
002752  104002                          HLT+2                   ;WRONG CONDITION CODES!

                        ;*****************************************************************
                        ;TEST 22:       TEST STCFI (STORE-CONVERT FLOATING TO INTEGER.
                        ;       STORE   043777,177777 --> 077777
                        ;       FPS = 047400,   AC = AC1,       DST = M6-R7
                        ;*****************************************************************

002754  104400                          SCOPE
002756  000402                          BR      TST22

002760  043777  177777          DAT22:  043777,177777

002764  170127  047400          TST22:  LDFPS   #047400         ;LOAD FLOATING POINT STATUS
002770  172567  177764                  LDF     DAT22,  AC1     ;LOAD 043777,177777 INTO AC1
002774  175567  176002          FPI22:  STCFI   AC1,    ANS1    ;STORE-CONVERT AC1 IN ANS1
003000  013767  177776  175776          MOV     @#PS,   ANS2    ;GET CPU STATUS
003006  042767  177760  175770          BIC     #177760,ANS2    ;SAVE CONDITION CODES
003014  170200                          STFPS   FPS             ;STORE FLOATING POINT STATUS
003016  022700  047400                  CMP     #047400,FPS     ;CHECK FLOATING POINT STATUS
003022  001401                          BEQ     .+4             ;BRANCH IF OK
003024  104000                          HLT                     ;FPS NOT EQUAL TO 047400

003026  022767  077777  175746          CMP     #077777,ANS1    ;DID 077777 GET STORED?
003034  001401                          BEQ     .+4             ;BRANCH IF OK
003036  104001                          HLT+1                   ;ANS1 NOT EQUAL TO 077777

003040  022767  000000  175736          CMP     #0,     ANS2    ;CONDITION CODES = 0?
003046  001401                          BEQ     .+4             ;BRANCH IF OK
003050  104002                          HLT+2                   ;WRONG CONDITION CODES!

                        ;*****************************************************************
                        ;TEST 23:       TEST STCFI (STORE-CONVERT FLOATING TO INTEGER.
                        ;       STORE   044000,000000 --> 000000
                        ;       FPS = 147405,   AC = AC0,       DST = M6-R7
                        ;       FEC = 6,        FEA = FPI23
                        ;*****************************************************************

003052  104400                          SCOPE
003054  000402                          BR      TST23

003056  044000  000000          DAT23:  044000,000000

003062  170127  047400          TST23:  LDFPS   #047400         ;LOAD FLOATING POINT STATUS
003066  172467  177764                  LDF     DAT23,  AC0     ;LOAD 044000,000000 INTO AC0
003072  175467  175704          FPI23:  STCFI   AC0,    ANS1    ;STORE-CONVERT AC0 IN ANS1
003076  013767  177776  175700          MOV     @#PS,   ANS2    ;GET CPU STATUS
003104  042767  177760  175672          BIC     #177760,ANS2    ;SAVE CONDITION CODES
003112  170200                          STFPS   FPS             ;STORE FLOATING POINT STATUS
```

```
003114  170367  175702          STST    FEC                     ;STORE EXCEPTION CODES
003120  022700  147405          CMP     #147405,FPS             ;CHECK FLOATING POINT STATUS
003124  001401                  BEQ     .+4                     ;BRANCH IF OK
003126  104000                  HLT                             ;FPS NOT EQUAL TO 147405

003130  022767  000006  175664  CMP     #6,     FEC             ;CHECK FLOATING EXCEPTION CODE
003136  001401                  BEQ     .+4                     ;BRANCH IF OK
003140  104000                  HLT                             ;FEC NOT EQUAL TO 6

003142  022767  003072  175654  CMP     #FPI23, FEA             ;CHECK FLOATING EXCEPTION ADDRESS
003150  001401                  BEQ     .+4                     ;BRANCH IF OK
003152  104000                  HLT                             ;FEA NOT EQUAL TO FPI23

003154  022767  000000  175620  CMP     #000000,ANS1            ;DID 000000 GET STORED?
003162  001401                  BEQ     .+4                     ;BRANCH IF OK
003164  104001                  HLT+1                           ;ANS1 NOT EQUAL TO 000000

003166  022767  000005  175610  CMP     #5,     ANS2            ;CONDITION CODES = 5?
003174  001401                  BEQ     .+4                     ;BRANCH IF OK
003176  104002                  HLT+2                           ;WRONG CONDITION CODES!
```

```
;*****************************************************************************
;TEST 24:       TEST STCFI (STORE-CONVERT FLOATING TO INTEGER)
;       STORE   144000,000377 --> 100000
;       FPS = 047410,   AC = AC0,       DST = M6-R7
;*****************************************************************************

003200  104400                  SCOPE
003202  000402                  BR      TST24

003204  144000  000377          DAT24:  144000,000377

003210  170127  047400          TST24:  LDFPS   #047400         ;LOAD FLOATING POINT STATUS
003214  172467  177764                  LDF     DAT24,  AC0     ;LOAD 144000,000377 INTO AC0
003220  175467  175556          FPI24:  STCFI   AC0,    ANS1    ;STORE-CONVERT AC0 IN ANS1
003224  013767  177776  175552          MOV     @#PS,   ANS2    ;GET CPU STATUS
003232  042767  177760  175544          BIC     #177760,ANS2    ;SAVE CONDITION CODES
003240  170200                  STFPS   FPS                     ;STORE FLOATING POINT STATUS
003242  022700  047410                  CMP     #047410,FPS     ;CHECK FLOATING POINT STATUS
003246  001401                  BEQ     .+4                     ;BRANCH IF OK
003250  104000                  HLT                             ;FPS NOT EQUAL TO 047410

003252  022767  100000  175522          CMP     #100000,ANS1    ;DID 100000 GET STORED?
003260  001401                  BEQ     .+4                     ;BRANCH IF OK
003262  104001                  HLT+1                           ;ANS1 NOT EQUAL TO 100000

003264  022767  000010  175512          CMP     #10,    ANS2    ;CONDITION CODES = 10?
003272  001401                  BEQ     .+4                     ;BRANCH IF OK
003274  104002                  HLT+2                           ;WRONG CONDITION CODES!
```

```
;*****************************************************************************
;TEST 25:       TEST STCFI (STORE-CONVERT FLOATING TO INTEGER
;       STORE   144000,000400 --> 000000
;       FPS = 147405,   AC = AC3,       DST = M6-R7
```

```
                        ;       FEC = 6.        FEA = FPI25
                        ;*******************************************************************************

003276  104400                  SCOPE
003300  000402                  BR      TST25

003302  144000  000400  DAT25:  144000,000400

003306  170127  047400  TST25:  LDFPS   #047400          ;LOAD FLOATING POINT STATUS
003312  172767  177764          LDF     DAT25,  AC3     ;LOAD 144000,000400 INTO AC3
003316  175767  175460  FPI25:  STCFI   AC3,    ANS1    ;STCFI-CONVERT AC3 IN ANS1
003322  013767  177776  175454  MOV     @#PS,   ANS2    ;GET CPU STATUS
003330  042767  177760  175446  BIC     #177760,ANS2    ;SAVE CONDITION CODES
003336  170200                  STFPS   FPS             ;STORE FLOATING POINT STATUS
003340  170367  175456          STST    FEC             ;STORE EXCEPTION CODES
003344  022700  147405          CMP     #147405,FPS     ;CHECK FLOATING POINT STATUS
003350  001401                  BEQ     .+4             ;BRANCH IF OK
003352  104000                  HLT                     ;FPS NOT EQUAL TO 147405

003354  022767  000006  175440  CMP     #6.     FEC     ;CHECK FLOATING EXCEPTION CODE
003362  001401                  BEQ     .+4             ;BRANCH IF OK
003364  104000                  HLT                     ;FEC NOT EQUAL TO 6

003366  022767  003316  175430  CMP     #FPI25, FEA     ;CHECK FLOATING EXCEPTION ADDRESS
003374  001401                  BEQ     .+4             ;BRANCH IF OK
003376  104000                  HLT                     ;FEA NOT EQUAL TO FPI25

003400  022767  000000  175374  CMP     #000000,ANS1    ;DID 000000 GET STORED?
003406  001401                  BEQ     .+4             ;BRANCH IF OK
003410  104001                  HLT+1                   ;ANS1 NOT EQUAL TO 000000

003412  022767  000005  175364  CMP     #5,     ANS2    ;CONDITION CODES = 5?
003420  001401                  BEQ     .+4             ;BRANCH IF OK
003422  104002                  HLT+2                   ;WRONG CONDITION CODES!


                        ;*******************************************************************************
                        ;TEST 26:       TEST STCFI (STORE-CONVERT FLOATING TO INTEGER)
                        ;       STORE   100000,000000 --> 000000
                        ;       FPS = 047404,   AC = AC3,       DST = M6-R7
                        ;*******************************************************************************

003424  104400                  SCOPE
003426  000402                  BR      TST26

003430  100000  000000  DAT26:  100000,000000

003434  170127  040000  TST26:  LDFPS   #040000          ;LOAD FLOATING POINT STATUS
003440  172767  177764          LDF     DAT26,  AC3     ;LOAD 100000,000000 INTO AC3
003444  170127  047400          LDFPS   #047400         ;LOAD FLOATING POINT STATUS
003450  175767  175326          STCFI   AC3,    ANS1    ;STORE-CONVERT AC3 IN ANS1
003454  013767  177776  175322  MOV     @#PS,   ANS2    ;GET CPU STATUS
003462  042767  177760  175314  BIC     #177760,ANS2    ;SAVE CONDITION CODES
003470  170200                  STFPS   FPS             ;STORE FLOATING POINT STATUS
003472  022700  047404          CMP     #047404,FPS     ;CHECK FLOATING POINT STATUS
003476  001401                  BEQ     .+4             ;BRANCH IF OK
```

```
003500  104000                          HLT                     ;FPS NOT EQUAL TO 047404

003502  022767  000000  175272          CMP     #000000,ANS1    ;DID 000000 GET STORED?
003510  001401                          BEQ     .+4             ;BRANCH IF OK
003512  104001                          HLT+1                   ;ANS1 NOT EQUAL TO 000000

003514  022767  000004  175262          CMP     #4,     ANS2    ;CONDITION CODES = 4?
003522  001401                          BEQ     .+4             ;BRANCH IF OK
003524  104002                          HLT+2                   ;WRONG CONDITION CODES!
```

```
                        ;*************************************************************************
                        ;TEST 27:       TEST STCFI (STORE-CONVERT FLOATING TO INTEGER)
                        ;       STORE   177777,177777 --> 000000
                        ;       FPS = 147405,   AC = AC0,       DST = M0-R1
                        ;       FEC = 6,        FEA = FPI27
                        ;*************************************************************************

003526  104400                  SCOPE
003530  000402                  BR      TST27

003532  177777  177777  DAT27:  177777,177777

003536  170127  047400  TST27:  LDFPS   #047400          ;LOAD FLOATING POINT STATUS
003542  172467  177764          LDF     DAT27,  AC0     ;LOAD 177777,177777 INTO AC0
003546  175401          FPI27:  STCFI   AC0,    R1      ;STORE-CONVERT AC0 IN R1
003550  013767  177776  175226  MOV     @#PS,   ANS2    ;GET CPU STATUS
003556  042767  177760  175220  BIC     #177760,ANS2    ;SAVE CONDITION CODES
003564  010167  175212          MOV     R1,     ANS1    ;SAVE R1
003570  170200                  STFPS   FPS             ;STORE FLOATING POINT STATUS
003572  170367  175224          STST    FEC             ;STORE EXCEPTION CODES
003576  022700  147405          CMP     #147405,FPS     ;CHECK FLOATING POINT STATUS
003602  001401                  BEQ     .+4             ;BRANCH IF OK
003604  104000                  HLT                     ;FPS NOT EQUAL TO 147405

003606  022767  000006  175206  CMP     #6,     FEC     ;CHECK FLOATING EXCEPTION CODE
003614  001401                  BEQ     .+4             ;BRANCH IF OK
003616  104000                  HLT                     ;FEC NOT EQUAL TO 6

003620  022767  003546  175176  CMP     #FPI27, FEA     ;CHECK FLOATING EXCEPTION ADDRESS
003626  001401                  BEQ     .+4             ;BRANCH IF OK
003630  104000                  HLT                     ;FEA NOT EQUAL TO FPI27

003632  022767  000000  175142  CMP     #000000,ANS1    ;DID 000000 GET STORED?
003640  001401                  BEQ     .+4             ;BRANCH IF OK
003642  104001                  HLT+1                   ;ANS1 NOT EQUAL TO 000000

003644  022767  000005  175132  CMP     #5,     ANS2    ;CONDITION CODES = 5?
003652  001401                  BEQ     .+4             ;BRANCH IF OK
003654  104002                  HLT+2                   ;WRONG CONDITION CODES!


                        ;*************************************************************************
                        ;TEST 30:       TEST STCFI (STORE-CONVERT FLOATING TO INTEGER)
                        ;       STORE   000200,000000 --> 000000
                        ;       FPS = 047404,   AC = AC2,       DST = M2-R7
                        ;*************************************************************************

003656  104400                  SCOPE
003660  000402                  BR      TST30

003662  000200  000000  DAT30:  000200,000000

003666  170127  047400  TST30:  LDFPS   #047400          ;LOAD FLOATING POINT STATUS
003672  172667  177764          LDF     DAT30,  AC2     ;LOAD 000200,000000 INTO AC2
003676  175627                  STCFI   AC2,    (PC)+   ;STORE-CONVERT AC2 IN NEXT LOC
```

```
003700  000000                  ANR30:  0
003702  000402                          BR      .+6             ;SKIP HALTS
003704  000000                          HALT                    ;PC FAILURE
003706  000000                          HALT                    ;
003710  013767  177776  175066          MOV     @#PS,   ANS2    ;GET CPU STATJS
003716  042767  177760  175060          BIC     #177760,ANS2    ;SAVE CONDITION CODES
003724  016767  177750  175050          MOV     ANR30,  ANS1    ;SAVE THE ANSWER
003732  170200                          STFPS   FPS             ;STORE FLOATING POINT STATUS
003734  022700  047404                  CMP     #047404,FPS     ;CHECK FLOATING POINT STATJS
003740  001401                          BEQ     .+4             ;BRANCH IF OK
003742  104000                          HLT                     ;FPS NOT EQUAL TO 047404

003744  022767  000000  175030          CMP     #000000,ANS1    ;DID 000000 GET STORED?
003752  001401                          BEQ     .+4             ;BRANCH IF OK
003754  104001                          HLT+1                   ;ANS1 NOT EQUAL TO 000000

003756  022767  000004  175020          CMP     #4,     ANS2    ;CONDITION CODES = 4?
003764  001401                          BEQ     .+4             ;BRANCH IF OK
003766  104002                          HLT+2                   ;WRONG CONDITION CODES!
```

```
;*****************************************************************
;TEST 31:       TEST STCFI (STORE-CONVERT FLOATING TO INTEGER)
;       STORE   100177,177777 --> 000000
;       FPS = 047404,   AC = AC1,       DST = M6-R7
;*****************************************************************

003770  104400                          SCOPE
003772  000402                          BR      TST31

003774  100177  177777          DAT31:  100177,177777

004000  170127  040000          TST31:  LDFPS   #040000         ;LOAD FLOATING POINT STATUS
004004  172567  177764                  LDF     DAT31,  AC1     ;LOAD 100177,177777 INTO AC1
004010  170127  047400                  LDFPS   #047400         ;LOAD FLOATING POINT STATJS
004014  175567  174762                  STCFI   AC1,    ANS1    ;STORE-CONVERT AC1 IN ANS1
004020  013767  177776  174756          MOV     @#PS,   ANS2    ;GET CPU STATUS
004026  042767  177760  174750          BIC     #177760,ANS2    ;SAVE CONDITION CODES
004034  170200                          STFPS   FPS             ;STORE FLOATING POINT STATUS
004036  022700  047404                  CMP     #047404,FPS     ;CHECK FLOATING POINT STATUS
004042  001401                          BEQ     .+4             ;BRANCH IF OK
004044  104000                          HLT                     ;FPS NOT EQUAL TO 047404

004046  022767  000000  174726          CMP     #000000,ANS1    ;DID 000000 GET STORED?
004054  001401                          BEQ     .+4             ;BRANCH IF OK
004056  104001                          HLT+1                   ;ANS1 NOT EQUAL TO 000000

004060  022767  000004  174716          CMP     #4,     ANS2    ;CONDITION CODES = 4?
004066  001401                          BEQ     .+4             ;BRANCH IF OK
004070  104002                          HLT+2                   ;WRONG CONDITION CODES!
```

```
;*****************************************************************
;TEST 32:       TEST LDCID (LOAD-CONVERT INTEGER TO DOUBLE
;       LOAD    000000 --> 000000,000000,000000,000000
;       FPS = 047604,   SRC = M2-R7,    AC = AC0
```

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

004072  104400                            SCOPE
004074  170127  047600          TST32:    LDFPS   #047600         ;LOAD FLOATING POINT STATUS
004100  177027  000000                    LDCID   #000000,AC0     ;LOAD-CONVERT 000000 INTO AC0
004104  170200                            STFPS   FPS             ;STORE FLOATING POINT STATUS
004106  022700  047604                    CMP     #047604,FPS     ;CHECK FLOATING POINT STATUS
004112  001401                            BEQ     .+4             ;BRANCH IF OK
004114  104000                            HLT                     ;FPS NOT EQUAL TO 047604

004116  174067  174660                    STD     AC0,    ANS1    ;STORE AC0 IN ANS1 THRU ANS4
004122  022767  000000  174652            CMP     #000000,ANS1    ;DID 000000 GET STORED?
004130  001401                            BEQ     .+4             ;BRANCH IF OK
004132  104004                            HLT+4                   ;ANS1 NOT EQUAL TO 000000

004134  022767  000000  174642            CMP     #000000,ANS2    ;DID 000000 GET STORED?
004142  001401                            BEQ     .+4             ;BRANCH IF OK
004144  104004                            HLT+4                   ;ANS2 NOT EQUAL TO 000000

004146  022767  000000  174632            CMP     #000000,ANS3    ;DID 000000 GET STORED?
004154  001401                            BEQ     .+4             ;BRANCH IF OK
004156  104004                            HLT+4                   ;ANS3 NOT EQUAL TO 000000

004160  022767  000000  174622            CMP     #000000,ANS4    ;DID 000000 GET STORED?
004166  001401                            BEQ     .+4             ;BRANCH IF OK
004170  104004                            HLT+4                   ;ANS4 NOT EQUAL TO 000000

                                ;**********************************************************************
                                ;TEST 33:       TEST LDCID (LOAD-CONVERT INTEGER TO DOUBLE)
                                ;       LOAD    125252 --> 143652,126000,000000,000000
                                ;       FPS = 047610,   SRC = M2-R7,    AC = AC0
                                ;**********************************************************************

004172  104400                            SCOPE
004174  170127  047600          TST33:    LDFPS   #047600         ;LOAD FLOATING POINT STATUS
004200  177027  125252                    LDCID   #125252,AC0     ;LOAD-CONVERT 125252 INTO AC0
004204  170200                            STFPS   FPS             ;STORE FLOATING POINT STATUS
004206  022700  047610                    CMP     #047610,FPS     ;CHECK FLOATING POINT STATUS
004212  001401                            BEQ     .+4             ;BRANCH IF OK
004214  104000                            HLT                     ;FPS NOT EQUAL TO 047610

004216  174067  174560                    STD     AC0,    ANS1    ;STORE AC0 IN ANS1 THRU ANS4
004222  022767  143652  174552            CMP     #143652,ANS1    ;DID 143652 GET STORED?
004230  001401                            BEQ     .+4             ;BRANCH IF OK
004232  104004                            HLT+4                   ;ANS1 NOT EQUAL TO 143652

004234  022767  126000  174542            CMP     #126000,ANS2    ;DID 126000 GET STORED?
004242  001401                            BEQ     .+4             ;BRANCH IF OK
004244  104004                            HLT+4                   ;ANS2 NOT EQUAL TO 126000

004246  022767  000000  174532            CMP     #000000,ANS3    ;DID 000000 GET STORED?
004254  001401                            BEQ     .+4             ;BRANCH IF OK
004256  104004                            HLT+4                   ;ANS3 NOT EQUAL TO 000000

004260  022767  000000  174522            CMP     #000000,ANS4    ;DID 000000 GET STORED?
```

```
004266  001401                          BEQ     .+4             ;BRANCH IF OK
004270  104004                          HLT+4                   ;ANS4 NOT EQUAL TO 000000


                        ;*****************************************************************
                        ;TEST 34:       TEST LDCID (LOAD-CONVERT INTEGER TO DOUBLE)
                        ;       LOAD    052525 --> 043652,125000,000000,000000
                        ;       FPS = 047600,   SRC = M2-R7,    AC = AC3
                        ;*****************************************************************

004272  104400                          SCOPE
004274  170127  047600          TST34:  LDFPS   #047600         ;LOAD FLOATING POINT STATUS
004300  177327  052525                  LDCID   #052525,AC3     ;LOAD-CONVERT 052525 INTO AC3
004304  170200                          STFPS   FPS             ;STORE FLOATING POINT STATUS
004306  022700  047600                  CMP     #047600,FPS     ;CHECK FLOATING POINT STATUS
004312  001401                          BEQ     .+4             ;BRANCH IF OK
004314  104000                          HLT                     ;FPS NOT EQUAL TO 047600

004316  174367  174460                  STD     AC3,    ANS1    ;STORE AC3 IN ANS1 THRU ANS4
004322  022767  043652  174452          CMP     #043652,ANS1    ;DID 043652 GET STORED?
004330  001401                          BEQ     .+4             ;BRANCH IF OK
004332  104004                          HLT+4                   ;ANS1 NOT EQUAL TO 043652

004334  022767  125000  174442          CMP     #125000,ANS2    ;DID 125000 GET STORED?
004342  001401                          BEQ     .+4             ;BRANCH IF OK
004344  104004                          HLT+4                   ;ANS2 NOT EQUAL TO 125000

004346  022767  000000  174432          CMP     #000000,ANS3    ;DID 000000 GET STORED?
004354  001401                          BEQ     .+4             ;BRANCH IF OK
004356  104004                          HLT+4                   ;ANS3 NOT EQUAL TO 000000

004360  022767  000000  174422          CMP     #000000,ANS4    ;DID 000000 GET STORED?
004366  001401                          BEQ     .+4             ;BRANCH IF OK
004370  104004                          HLT+4                   ;ANS4 NOT EQUAL TO 000000


                        ;*****************************************************************
                        ;TEST 35:       TEST LDCID (LOAD-CONVERT INTEGER TO DOUBLE)
                        ;       LOAD    000001 --> 040200,000000,000000,000000
                        ;       FPS = 047600,   SRC = M2-R7,    AC = AC1
                        ;*****************************************************************

004372  104400                          SCOPE
004374  170127  047600          TST35:  LDFPS   #047600         ;LOAD FLOATING POINT STATUS
004400  177127  000001                  LDCID   #000001,AC1     ;LOAD-CONVERT 000001 INTO AC1
004404  170200                          STFPS   FPS             ;STORE FLOATING POINT STATUS
004406  022700  047600                  CMP     #047600,FPS     ;CHECK FLOATING POINT STATUS
004412  001401                          BEQ     .+4             ;BRANCH IF OK
004414  104000                          HLT                     ;FPS NOT EQUAL TO 047600

004416  174167  174360                  STD     AC1,    ANS1    ;STORE AC1 IN ANS1 THRU ANS4
004422  022767  040200  174352          CMP     #040200,ANS1    ;DID 040200 GET STORED?
004430  001401                          BEQ     .+4             ;BRANCH IF OK
004432  104004                          HLT+4                   ;ANS1 NOT EQUAL TO 040200

004434  022767  000000  174342          CMP     #000000,ANS2    ;DID 000000 GET STORED?
```

```
                                        TEST SECTION

004442  001401                        BEQ     .+4                 ;BRANCH IF OK
004444  104004                        HLT+4                       ;ANS2 NOT EQUAL TO 000000

004446  022767  000000  174332        CMP     #000000,ANS3        ;DID 000000 GET STORED?
004454  001401                        BEQ     .+4                 ;BRANCH IF OK
004456  104004                        HLT+4                       ;ANS3 NOT EQUAL TO 000000

004460  022767  000000  174322        CMP     #000000,ANS4        ;DID 000000 GET STORED?
004466  001401                        BEQ     .+4                 ;BRANCH IF OK
004470  104004                        HLT+4                       ;ANS4 NOT EQUAL TO 000000


                        ;*****************************************************************
                        ;TEST 36:       TEST LDCID (LOAD-CONVERT INTEGER TO DOUBLE)
                        ;       LOAD    177777 --> 140200,000000,000000,000000
                        ;       FPS = 047610,   SRC = M2-R7,    AC = AC1
                        ;*****************************************************************

004472  104400                        SCOPE
004474  170127  047600          TST36: LDFPS   #047600             ;LOAD FLOATING POINT STATUS
004500  177127  177777                 LDCID   #177777,AC1         ;LOAD-CONVERT 177777 INTO AC1
004504  170200                        STFPS   FPS                 ;STORE FLOATING POINT STATUS
004506  022700  047610                CMP     #047610,FPS         ;CHECK FLOATING POINT STATUS
004512  001401                        BEQ     .+4                 ;BRANCH IF OK
004514  104000                        HLT                         ;FPS NOT EQUAL TO 047610
004516  174167  174260                STD     AC1,    ANS1        ;STORE AC1 IN ANS1 THRU ANS4
004522  022767  140200  174252        CMP     #140200,ANS1        ;DID 140200 GET STORED?
004530  001401                        BEQ     .+4                 ;BRANCH IF OK
004532  104004                        HLT+4                       ;ANS1 NOT EQUAL TO 140200

004534  022767  000000  174242        CMP     #000000,ANS2        ;DID 000000 GET STORED?
004542  001401                        BEQ     .+4                 ;BRANCH IF OK
004544  104004                        HLT+4                       ;ANS2 NOT EQUAL TO 000000

004546  022767  000000  174232        CMP     #000000,ANS3        ;DID 000000 GET STORED?
004554  001401                        BEQ     .+4                 ;BRANCH IF OK
004556  104004                        HLT+4                       ;ANS3 NOT EQUAL TO 000000

004560  022767  000000  174222        CMP     #000000,ANS4        ;DID 000000 GET STORED?
004566  001401                        BEQ     .+4                 ;BRANCH IF OK
004570  104004                        HLT+4                       ;ANS4 NOT EQUAL TO 000000


                        ;*****************************************************************
                        ;TEST 37:       TEST LDCID (LOAD-CONVERT INTEGER TO DOUBLE)
                        ;       LOAD    100000 --> 144000,000000,000000,000000
                        ;       FPS = 047610,   SRC = M2-R7,    AC = AC1
                        ;*****************************************************************

004572  104400                        SCOPE
004574  170127  047600          TST37: LDFPS   #047600             ;LOAD FLOATING POINT STATUS
004600  177127  100000                 LDCID   #100000,AC1         ;LOAD-CONVERT 100000 INTO AC1
004604  170200                        STFPS   FPS                 ;STORE FLOATING POINT STATUS
004606  022700  047610                CMP     #047610,FPS         ;CHECK FLOATING POINT STATUS
004612  001401                        BEQ     .+4                 ;BRANCH IF OK
```

```
004614  104000                          HLT                     ;FPS NOT EQUAL TO 047610

004616  174167  174160                  STD     AC1,    ANS1    ;STORE AC1 IN ANS1 THRU ANS4
004622  022767  144000  174152          CMP     #144000,ANS1    ;DID 144000 GET STORED?
004630  001401                          BEQ     .+4             ;BRANCH IF OK
004632  104004                          HLT+4                   ;ANS1 NOT EQUAL TO 144000

004634  022767  000000  174142          CMP     #000000,ANS2    ;DID 000000 GET STORED?
004642  001401                          BEQ     .+4             ;BRANCH IF OK
004644  104004                          HLT+4                   ;ANS2 NOT EQUAL TO 000000

004646  022767  000000  174132          CMP     #000000,ANS3    ;DID 000000 GET STORED?
004654  001401                          BEQ     .+4             ;BRANCH IF OK
004656  104004                          HLT+4                   ;ANS3 NOT EQUAL TO 000000

004660  022767  000000  174122          CMP     #000000,ANS4    ;DID 000000 GET STORED?
004666  001401                          BEQ     .+4             ;BRANCH IF OK
004670  104004                          HLT+4                   ;ANS4 NOT EQUAL TO 000000
```

```
;*****************************************************************************
;TEST 40:       TEST LDCID (LOAD-CONVERT INTEGER TO DOUBLE)
;       LOAD    077777 --> 043777,177000,000000,000000
;       FPS = 047600,   SRC = MC-R3,    AC = AC2
;*****************************************************************************

004672  104400                          SCOPE
004674  170127  047600          TST40:  LDFPS   #047600         ;LOAD FLOATING POINT STATUS
004700  012703  077777                  MOV     #077777,R3      ;LOAD 077777 INTO R3
004704  177203                          LDCID   R3,     AC2     ;LOAD-CONVERT 077777 INTO AC2
004706  170200                          STFPS   FPS             ;STORE FLOATING POINT STATUS
004710  022700  047600                  CMP     #047600,FPS     ;CHECK FLOATING POINT STATUS
004714  001401                          BEQ     .+4             ;BRANCH IF OK
004716  104000                          HLT                     ;FPS NOT EQUAL TO 047600

004720  174267  174056                  STD     AC2,    ANS1    ;STORE AC2 IN ANS1 THRU ANS4
004724  022767  043777  174050          CMP     #043777,ANS1    ;DID 043777 GET STORED?
004732  001401                          BEQ     .+4             ;BRANCH IF OK
004734  104004                          HLT+4                   ;ANS1 NOT EQUAL TO 043777

004736  022767  177000  174040          CMP     #177000,ANS2    ;DID 177000 GET STORED?
004744  001401                          BEQ     .+4             ;BRANCH IF OK
004746  104004                          HLT+4                   ;ANS2 NOT EQUAL TO 177000

004750  022767  000000  174030          CMP     #000000,ANS3    ;DID 000000 GET STORED?
004756  001401                          BEQ     .+4             ;BRANCH IF OK
004760  104004                          HLT+4                   ;ANS3 NOT EQUAL TO 000000

004762  022767  000000  174020          CMP     #000000,ANS4    ;DID 000000 GET STORED?
004770  001401                          BEQ     .+4             ;BRANCH IF Ok
004772  104004                          HLT+4                   ;ANS4 NOT EQUAL TO 000000
```

```
;*****************************************************************************
;TEST 41:       TEST LDCID (LOAD-CONVERT INTEGER TO DOUBLE)
;       LOAD    000125 --> 041652,000000,000000,000000
```

```
                                ;       FPS = 047600,   SRC = M6-R7,    AC = AC1
                                ;*****************************************************************************

005774  104400                        SCOPE
005776  000481                        BR      TST41

005000  000125                DAT41:  000125

005002  170127  047600        TST41:  LDFPS   #047600         ;LOAD FLOATING POINT STATUS
005006  177167  177766                LDCID   DAT41,  AC1     ;LOAD-CONVERT 000125 INTO AC1
005012  170200                        STFPS   FPS             ;STORE FLOATING POINT STATUS
005014  022700  047600                CMP     #047600,FPS     ;CHECK FLOATING POINT STATUS
005020  001401                        BEQ     .+4             ;BRANCH IF OK
005022  104000                        HLT                     ;FPS NOT EQUAL TO 047600

005024  174167  173752                STD     AC1,    ANS1    ;STORE AC1 IN ANS1 THRU ANS4
005030  022767  041652  173744        CMP     #041652,ANS1    ;DID 041652 GET STORED?
005036  001401                        BEQ     .+4             ;BRANCH IF OK
005040  104004                        HLT+4                   ;ANS1 NOT EQUAL TO 041652

005042  022767  000000  173734        CMP     #000000,ANS2    ;DID 000000 GET STORED?
005050  001401                        BEQ     .+4             ;BRANCH IF OK
005052  104004                        HLT+4                   ;ANS2 NOT EQUAL TO 000000

005054  022767  000000  173724        CMP     #000000,ANS3    ;DID 000000 GET STORED?
005062  001401                        BEQ     .+4             ;BRANCH IF OK
005064  104004                        HLT+4                   ;ANS3 NOT EQUAL TO 000000

005066  022767  000000  173714        CMP     #000000,ANS4    ;DID 000000 GET STORED?
005074  001401                        BEQ     .+4             ;BRANCH IF OK
005076  104004                        HLT+4                   ;ANS4 NOT EQUAL TO 000000


                                ;*****************************************************************************
                                ;TEST 42:       TEST LDCID (LOAD-CONVERT INTEGER TO DOUBLE.
                                ;       LOAD    070707 --> 043743,107000,000000,000000
                                ;       FPS = 047600,   SRC = M2-R7,    AC = AC1
                                ;*****************************************************************************

005100  104400                        SCOPE
005102  170127  047600        TST42:  LDFPS   #047600         ;LOAD FLOATING POINT STATUS
005106  177127  070707                LDCID   #070707,AC1     ;LOAD-CONVERT 070707 INTO AC1
005112  170200                        STFPS   FPS             ;STORE FLOATING POINT STATUS
005114  022700  047600                CMP     #047600,FPS     ;CHECK FLOATING POINT STATUS
005120  001401                        BEQ     .+4             ;BRANCH IF OK
005122  104000                        HLT                     ;FPS NOT EQUAL TO 047600

005124  174167  173652                STD     AC1,    ANS1    ;STORE AC1 IN ANS1 THRU ANS4
005130  022767  043743  173644        CMP     #043743,ANS1    ;DID 043743 GET STORED?
005136  001401                        BEQ     .+4             ;BRANCH IF OK
005140  104004                        HLT+4                   ;ANS1 NOT EQUAL TO 043743

005142  022767  107000  173634        CMP     #107000,ANS2    ;DID 107000 GET STORED?
005150  001401                        BEQ     .+4             ;BRANCH IF OK
005152  104004                        HLT+4                   ;ANS2 NOT EQUAL TO 107000
```

```
005154  022767  000000  173624          CMP     #000000,ANS3    ;DID 000000 GET STORED?
005162  001401                          BEQ     .+4             ;BRANCH IF OK
005164  104004                          HLT+4                   ;ANS3 NOT EQUAL TO 000000

005166  022767  000000  173614          CMP     #000000,ANS4    ;DID 000000 GET STORED?
005174  001401                          BEQ     .+4             ;BRANCH IF OK
005176  104004                          HLT+4                   ;ANS4 NOT EQUAL TO 000000


                                ;*****************************************************************
                                ;TEST 43:       TEST LDCID (LOAD-CONVERT INTEGER TO DOUBLE)
                                ;       LOAD    107070 --> 143743,110000,000000,000000
                                ;       FPS = 047610,   SRC = M2-R7,    AC = AC0
                                ;*****************************************************************

005200  104400                          SCOPE
005202  170127  047600          TST43:  LDFPS   #047600         ;LOAD FLOATING POINT STATUS
005206  177027  107070                  LDCID   #107070,AC0     ;LOAD-CONVERT 107070 INTO AC0
005212  170200                          STFPS   FPS             ;STORE FLOATING POINT STATUS
005214  022700  047610                  CMP     #047610,FPS     ;CHECK FLOATING POINT STATUS
005220  001401                          BEQ     .+4             ;BRANCH IF OK
005222  104000                          HLT                     ;FPS NOT EQUAL TO 047610

005224  174067  173552                  STD     AC0,    ANS1    ;STORE AC0 IN ANS1 THRU ANS4
005230  022767  143743  173544          CMP     #143743,ANS1    ;DID 143743 GET STORED?
005236  001401                          BEQ     .+4             ;BRANCH IF OK
005240  104004                          HLT+4                   ;ANS1 NOT EQUAL TO 143743

005242  022767  110000  173534          CMP     #110000,ANS2    ;DID 110000 GET STORED?
005250  001401                          BEQ     .+4             ;BRANCH IF OK
005252  104004                          HLT+4                   ;ANS2 NOT EQUAL TO 110000

005254  022767  000000  173524          CMP     #000000,ANS3    ;DID 000000 GET STORED?
005262  001401                          BEQ     .+4             ;BRANCH IF OK
005264  104004                          HLT+4                   ;ANS3 NOT EQUAL TO 000000

005266  022767  000000  173514          CMP     #000000,ANS4    ;DID 000000 GET STORED?
005274  001401                          BEQ     .+4             ;BRANCH IF OK
005276  104004                          HLT+4                   ;ANS4 NOT EQUAL TO 000000


                                ;*****************************************************************
                                ;TEST 44:       TEST STCDI (STORE-CONVERT DOUBLE TO INTEGER)
                                ;       STORE   000000,000000,000000,000000 --> 000000
                                ;       FPS = 047604,   AC = AC3,       DST = M6-R7
                                ;*****************************************************************

005300  104400                          SCOPE
005302  000424                          BR      TST44

005304  000000  000000  000000  DAT44:  000000,000000,000000,000000
005312  000000

005314  170127  047600          TST44:  LDFPS   #047600         ;LOAD FLOATING POINT STATUS
005320  172767  177760                  LDD     DAT44,  AC3     ;LOAD 000000,000000,000000,000000 INTO AC3
005324  175767  173452          FP144:  STCDI   AC3,    ANS1    ;STORE-CONVERT AC3 IN ANS1
```

```
005330  013767  177776  173446          MOV     @#PS,   ANS2    ;GET CPU STATUS
005336  042767  177760  173440          BIC     #177760,ANS2    ;SAVE CONDITION CODES
005344  170200                          STFPS   FPS             ;STORE FLOATING POINT STATUS
005346  022700  047604                  CMP     #047604,FPS     ;CHECK FLOATING POINT STATUS
005352  001401                          BEQ     .+4             ;BRANCH IF OK
005354  104000                          HLT                     ;FPS NOT EQUAL TO 047604

005356  022767  000000  173416          CMP     #000000,ANS1    ;DID 000000 GET STORED?
005364  001401                          BEQ     .+4             ;BRANCH IF OK
005366  104001                          HLT+1                   ;ANS1 NOT EQUAL TO 000000

005370  022767  000004  173406          CMP     #4,     ANS2    ;CONDITION CODES = 4?
005376  001401                          BEQ     .+4             ;BRANCH IF OK
005400  104002                          HLT+2                   ;WRONG CONDITION CODES!


                        ;*****************************************************************
                        ;TEST 45:      TEST STCDI (STORE-CONVERT DOUBLE TO INTEGER)
                        ;       STORE   041252,125252,125252,125252 --> 000025
                        ;       FPS = 047600,   AC = AC3,       DST = M6-R7
                        ;*****************************************************************

005402  104400                          SCOPE
005404  000404                          BR      TST45

005406  041252  125252  125252  DAT45:  041252,125252,125252,125252
005414  125252

005416  170127  047600          TST45:  LDFPS   #047600         ;LOAD FLOATING POINT STATUS
005422  172767  177760                  LDD     DAT45,  AC3     ;LOAD 041252,125252,125252,125252 INTO AC3
005426  175767  173350          FPI45:  STCDI   AC3,    ANS1    ;STORE-CONVERT AC3 IN ANS1
005432  013767  177776  173344          MOV     @#PS,   ANS2    ;GET CPU STATUS
005440  042767  177760  173336          BIC     #177760,ANS2    ;SAVE CONDITION CODES
005446  170200                          STFPS   FPS             ;STORE FLOATING POINT STATUS
005450  022700  047600                  CMP     #047600,FPS     ;CHECK FLOATING POINT STATUS
005454  001401                          BEQ     .+4             ;BRANCH IF OK
005456  104000                          HLT                     ;FPS NOT EQUAL TO 047600

005460  022767  000025  173314          CMP     #000025,ANS1    ;DID 000025 GET STORED?
005466  001401                          BEQ     .+4             ;BRANCH IF OK
005470  104001                          HLT+1                   ;ANS1 NOT EQUAL TO 000025

005472  022767  000000  173304          CMP     #0,     ANS2    ;CONDITION CODES = 0?
005500  001401                          BEQ     .+4             ;BRANCH IF OK
005502  104002                          HLT+2                   ;WRONG CONDITION CODES!


                        ;*****************************************************************
                        ;TEST 46:      TEST STCDI (STORE-CONVERT DOUBLE TO INTEGER)
                        ;       STORE   141252,125252,125252,125252 --> 177753
                        ;       FPS = 047610,   AC = AC2,       DST = M6-R7
                        ;*****************************************************************

005504  104400                          SCOPE
005506  000506                          BR      TST46
```

```
005510  141252  125252  125252  DAT46:  141252,125252,125252,125252
005516  125252

005520  170127  047600          TST46:  LDFPS   #047600          ;LOAD FLOATING POINT STATUS
005524  172667  177760                  LDD     DAT46,  AC2      ;LOAD 141252,125252,125252,125252 INTO AC2
005530  175667  173246          FPI46:  STCDI   AC2,    ANS1     ;STORE-CONVERT AC2 IN ANS1
005534  013767  177776  173242          MOV     @#PS,   ANS2     ;GET CPU STATUS
005542  042767  177760  173234          BIC     #177760,ANS2     ;SAVE CONDITION CODES
005550  170200                          STFPS   FPS              ;STORE FLOATING POINT STATUS
005552  022700  047610                  CMP     #047610,FPS      ;CHECK FLOATING POINT STATUS
005556  001401                          BEQ     .+4              ;BRANCH IF OK
005560  104000                          HLT                      ;FPS NOT EQUAL TO 047610

005562  022767  177753  173212          CMP     #177753,ANS1     ;DID 177753 GET STORED?
005570  001401                          BEQ     .+4              ;BRANCH IF OK
005572  104001                          HLT+1                    ;ANS1 NOT EQUAL TO 177753

005574  022767  000010  173202          CMP     #10.,   ANS2     ;CONDITION CODES = 10?
005602  001401                          BEQ     .+4              ;BRANCH IF OK
005604  104002                          HLT+2                    ;WRONG CONDITION CODES!

                        ;*****************************************************************
                        ;TEST 47:       TEST STCDI (STORE-CONVERT DOUBLE TO INTEGER
                        ;       STORE   040177,177777,177777,177777 --> 000000
                        ;       FPS = 047604,   AC = AC1,       DST = M6-R7
                        ;*****************************************************************

005606  104400                          SCOPE
005610  000404                          BR      TST47

005612  040177  177777  177777  DAT47:  040177,177777,177777,177777
005620  177777

005622  170127  047600          TST47:  LDFPS   #047600          ;LOAD FLOATING POINT STATUS
005626  172567  177760                  LDD     DAT47,  AC1      ;LOAD 040177,177777,177777,177777 INTO AC1
005632  175567  173144          FPI47:  STCDI   AC1,    ANS1     ;STORE-CONVERT AC1 IN ANS1
005636  013767  177776  173140          MOV     @#PS,   ANS2     ;GET CPU STATUS
005644  042767  177760  173132          BIC     #177760,ANS2     ;SAVE CONDITION CODES
005652  170200                          STFPS   FPS              ;STORE FLOATING POINT STATUS
005654  022700  047604                  CMP     #047604,FPS      ;CHECK FLOATING POINT STATUS
005660  001401                          BEQ     .+4              ;BRANCH IF OK
005662  104000                          HLT                      ;FPS NOT EQUAL TO 047604

005664  022767  000000  173110          CMP     #000000,ANS1     ;DID 000000 GET STORED?
005672  001401                          BEQ     .+4              ;BRANCH IF OK
005674  104001                          HLT+1                    ;ANS1 NOT EQUAL TO 000000

005676  022767  000004  173100          CMP     #4,     ANS2     ;CONDITION CODES = 4?
005704  001401                          BEQ     .+4              ;BRANCH IF OK
005706  104002                          HLT+2                    ;WRONG CONDITION CODES!

                        ;*****************************************************************
                        ;TEST 50:       TEST STCDI (STORE-CONVERT DOUBLE TO INTEGER.
                        ;       STORE   140177,177777,177777,177777 --> 000000
```

```
                        ;       FPS = 047604,   AC = AC0,       DST = M6-P7
                        ;*****************************************************************

005710  104400                    SCOPE
005712  000404                    BR      TST50

005714  140177  177777  177777  DAT50:  140177,177777,177777,177777
005722  177777

005724  170127  047600          TST50:  LDFPS   #047600         ;LOAD FLOATING POINT STATUS
005730  172467  177760                  LDD     DAT50,  AC0     ;LOAD 140177,177777,177777,177777 INTO AC0
005734  175467  173042          FPI50:  STCDI   AC0,    ANS1    ;STORE-CONVERT AC0 IN ANS1
005740  013767  177776  173036          MOV     @#PS,   ANS2    ;GET CPU STATUS
005746  042767  177760  173030          BIC     #177760,ANS2    ;SAVE CONDITION CODES
005754  170200                    STFPS   FPS                     ;STORE FLOATING POINT STATUS
005756  022700  047604                  CMP     #047604,FPS     ;CHECK FLOATING POINT STATUS
005762  001401                    BEQ     .+4                     ;BRANCH IF OK
005764  104000                    HLT                             ;FPS NOT EQUAL TO 047604

005766  022767  000000  173006          CMP     #000000,ANS1    ;DID 000000 GET STORED?
005774  001401                    BEQ     .+4                     ;BRANCH IF OK
005776  104001                    HLT+1                           ;ANS1 NOT EQUAL TO 000000

006000  022767  000004  172776          CMP     #4,     ANS2    ;CONDITION CODES = 4?
006006  001401                    BEQ     .+4                     ;BRANCH IF OK
006010  104002                    HLT+2                           ;WRONG CONDITION CODES!

                        ;*****************************************************************
                        ;TEST 51:       TEST STCDI (STORE-CONVERT DOUBLE TO INTEGER.
                        ;       STORE   040200,000000,000000,000001 --> 00001
                        ;       FPS = 047600,   AC = AC3,       DST = M6-P7
                        ;*****************************************************************

006012  104400                    SCOPE
006014  000404                    BR      TST51

006016  040200  000000  000000  DAT51:  040200,000000,000000,000001
006024  000001

006026  170127  047600          TST51:  LDFPS   #047600         ;LOAD FLOATING POINT STATUS
006032  172767  177750                  LDD     DAT51,  AC3     ;LOAD 040200,000000,000000,000001 INTO AC3
006036  175767  172740          FPI51:  STCDI   AC3,    ANS1    ;STORE-CONVERT AC3 IN ANS1
006042  013767  177776  172734          MOV     @#PS,   ANS2    ;GET CPU STATUS
006050  042767  177760  172726          BIC     #177760,ANS2    ;SAVE CONDITION CODES
006056  170200                    STFPS   FPS                     ;STORE FLOATING POINT STATUS
006060  022700  047600                  CMP     #047600,FPS     ;CHECK FLOATING POINT STATUS
006064  001401                    BEQ     .+4                     ;BRANCH IF OK
006066  104000                    HLT                             ;FPS NOT EQUAL TO 047600

006070  022767  000001  172734          CMP     #000001,ANS1    ;DID 000001 GET STORED?
006076  001401                    BEQ     .+4                     ;BRANCH IF OK
006100  104001                    HLT+1                           ;ANS1 NOT EQUAL TO 00001

006102  022767  000000  172674          CMP     #0,     ANS2    ;CONDITION CODES = 0?
006110  001401                    BEQ     .+4                     ;BRANCH IF OK
```

```
006112  104002                          HLT+2                   ;WRONG CONDITION CODES!


                        ;*****************************************************************
                        ;TEST 52:       TEST STCDI (STORE-CONVERT DOUBLE TO INTEGER)
                        ;       STORE   140200,000000,000000,000001 --> 177777
                        ;       FPS = 047610,   AC = AC3,       DST = M6-R7
                        ;*****************************************************************

006114  104400                          SCOPE
006116  000404                          BR      TST52

006120  140200  000000  000000  DAT52:  140200,000000,000000,000001
006126  000001

006130  170127  047600          TST52:  LDFPS   #047600         ;LOAD FLOATING POINT STATUS
006134  172767  177760                  LDD     DAT52,  AC3     ;LOAD 140200,000000,000000,000001 INTO AC3
006140  175767  172636          FPI52:  STCDI   AC3,    ANS1    ;STORE-CONVERT AC3 IN ANS1
006144  013767  177776  172632          MOV     @#PS,   ANS2    ;GET CPU STATUS
006152  042767  177760  172624          BIC     #177760,ANS2    ;SAVE CONDITION CODES
006160  170200                          STFPS   FPS             ;STORE FLOATING POINT STATUS
006162  022700  047610                  CMP     #047610,FPS     ;CHECK FLOATING POINT STATUS
006166  001401                          BEQ     .+4             ;BRANCH IF OK
006170  104000                          HLT                     ;FPS NOT EQUAL TO 047610

006172  022767  177777  172602          CMP     #177777,ANS1    ;DID 177777 GET STORED?
006200  001401                          BEQ     .+4             ;BRANCH IF OK
006202  104001                          HLT+1                   ;ANS1 NOT EQUAL TO 177777

006204  022767  000010  172572          CMP     #10,    ANS2    ;CONDITION CODES = 10?
006212  001401                          BEQ     .+4             ;BRANCH IF OK
006214  104002                          HLT+2                   ;WRONG CONDITION CODES!


                        ;*****************************************************************
                        ;TEST 53:       TEST STCDI (STORE-CONVERT DOUBLE TO INTEGER)
                        ;       STORE   043777,177777,177777,177777 --> 077777
                        ;       FPS = 047600,   AC = AC2,       DST = M6-R7
                        ;*****************************************************************

006216  104400                          SCOPE
006220  000404                          BR      TST53

006222  043777  177777  177777  DAT53:  043777,177777,177777,177777
006230  177777

006232  170127  047600          TST53:  LDFPS   #047600         ;LOAD FLOATING POINT STATUS
006236  172667  177760                  LDD     DAT53,  AC2     ;LOAD 043777,177777,177777,177777 INTO AC2
006242  175667  172534          FPI53:  STCDI   AC2,    ANS1    ;STORE-CONVERT AC2 IN ANS1
006246  013767  177776  172530          MOV     @#PS,   ANS2    ;GET CPU STATUS
006254  042767  177760  172522          BIC     #177760,ANS2    ;SAVE CONDITION CODES
006262  170200                          STFPS   FPS             ;STORE FLOATING POINT STATUS
006264  022700  047600                  CMP     #047600,FPS     ;CHECK FLOATING POINT STATUS
006270  001401                          BEQ     .+4             ;BRANCH IF OK
006272  104000                          HLT                     ;FPS NOT EQUAL TO 047600
```

```
006274  022767  077777  172500          CMP     #077777,ANS1    ;DID 077777 GET STORED?
006302  001401                          BEQ     .+4             ;BRANCH IF OK
006304  104001                          HLT+1                   ;ANS1 NOT EQUAL TO 077777

006306  022767  000000  172470          CMP     #0,     ANS2    ;CONDITION CODES = 0?
006314  001401                          BEQ     .+4             ;BRANCH IF OK
006316  104002                          HLT+2                   ;WRONG CONDITION CODES!


                                ;*****************************************************************
                                ;TEST 54:       TEST STCDI (STORE-CONVERT DOUBLE TO INTEGER)
                                ;       STORE   044000,000000,000000,000000 --> 000000
                                ;       FPS = 147605,   AC = AC1,       DST = M6-R7
                                ;       FEC = 6.        FEA = FPI54
                                ;*****************************************************************

006320  104400                          SCOPE
006322  000404                          BR      TST54

006324  044000  000000  000000  DAT54:  044000,000000,000000,000000
006332  000000

006334  170127  047600          TST54:  LDFPS   #047600         ;LOAD FLOATING POINT STATUS
006340  172567  177760                  LDD     DAT54,  AC1     ;LOAD 044000,000000,000000,000000 INTO AC1
006344  175567  172432          FPI54:  STCDI   AC1,    ANS1    ;STORE-CONVERT AC1 IN ANS1
006350  013767  177776  172426          MOV     @#PS,   ANS2    ;GET CPU STATUS
006356  042767  177760  172420          BIC     #177760,ANS2    ;SAVE CONDITION CODES
006364  170200                          STFPS   FPS             ;STORE FLOATING POINT STATUS
006366  170367  172430                  STST    FEC             ;STORE EXCEPTION CODES
006372  022700  147605                  CMP     #147605,FPS     ;CHECK FLOATING POINT STATUS
006376  001401                          BEQ     .+4             ;BRANCH IF OK
006400  104000                          HLT                     ;FPS NOT EQUAL TO 147605

006402  022767  000006  172412          CMP     #6,     FEC     ;CHECK FLOATING EXCEPTION CODE
006410  001401                          BEQ     .+4             ;BRANCH IF OK
006412  104000                          HLT                     ;FEC NOT EQUAL TO 6

006414  022767  006344  172402          CMP     #FPI54, FEA     ;CHECK FLOATING EXCEPTION ADDRESS
006422  001401                          BEQ     .+4             ;BRANCH IF OK
006424  104000                          HLT                     ;FEA NOT EQUAL TO FPI54

006426  022767  000000  172346          CMP     #000000,ANS1    ;DID 000000 GET STORED?
006434  001401                          BEQ     .+4             ;BRANCH IF OK
006436  104001                          HLT+1                   ;ANS1 NOT EQUAL TO 000000

006440  022767  000005  172336          CMP     #5,     ANS2    ;CONDITION CODES = 5?
006446  001401                          BEQ     .+4             ;BRANCH IF OK
006450  104002                          HLT+2                   ;WRONG CONDITION CODES!


                                ;*****************************************************************
                                ;TEST 55:       TEST STCDI (STORE-CONVERT DOUBLE TO INTEGER
                                ;       STORE   144000,000377,177777,177777 --> 100000
                                ;       FPS = 047610,   AC = AC3,       DST = M0-R5
                                ;*****************************************************************
```

```
006452  104400                          SCOPE
006454  000404                          BR      TST55

006456  144000  000377  177777  DAT55:  144000,000377,177777,177777
006464  177777

006466  170127  047600          TST55:  LDFPS   #047600         ;LOAD FLOATING POINT STATUS
006472  172767  177760                  LDD     DAT55,  AC3     ;LOAD 144000,000377,177777,177777 INTO AC3
006476  175705                          STCDI   AC3,    R5      ;STORE-CONVERT AC3 IN R5
006500  013767  177776  172276          MOV     @#PS,   ANS2    ;GET CPU STATUS
006506  042767  177760  172270          BIC     #177760,ANS2    ;SAVE CONDITION CODES
006514  010567  172262                  MOV     R5,     ANS1    ;SAVE R5
006520  170200                          STFPS   FPS             ;STORE FLOATING POINT STATUS
006522  022700  047610                  CMP     #047610,FPS     ;CHECK FLOATING POINT STATUS
006526  001401                          BEQ     .+4             ;BRANCH IF OK
006530  104000                          HLT                     ;FPS NOT EQUAL TO 047610

006532  022767  100000  172242          CMP     #100000,ANS1    ;DID 100000 GET STORED?
006540  001401                          BEQ     .+4             ;BRANCH IF OK
006542  104001                          HLT+1                   ;ANS1 NOT EQUAL TO 100000

006544  022767  000010  172232          CMP     #10.,   ANS2    ;CONDITION CODES = 10?
006552  001401                          BEQ     .+4             ;BRANCH IF OK
006554  104002                          HLT+2                   ;WRONG CONDITION CODES!

                        ;*****************************************************************
                        ;TEST 56:       TEST STCDI (STORE-CONVERT DOUBLE TO INTEGER)
                        ;       STORE   144000,000400,000000,000000 --> 000000
                        ;       FPS = 147605,   AC = AC2,       DST = M2-R7
                        ;       FEC = 6,        FEA = FPI56
                        ;*****************************************************************

006556  104400                          SCOPE
006560  000404                          BR      TST56

006562  144000  000400  000000  DAT56:  144000,000400,000000,000000
006570  000000

006572  170127  047600          TST56:  LDFPS   #047600         ;LOAD FLOATING POINT STATUS
006576  172667  177760                  LDD     DAT56,  AC2     ;LOAD 144000,000400,000000,000000 INTO AC2
006602  175627                  FPI56:  STCDI   AC2,    (PC)+   ;STORE-CONVERT AC2 IN ANS1
006604  000000                  ANR56:  0                       ;LOCATION FOR ANS
006606  000402                          BR      .+6
006610  000000                          HALT                    ;PC FAILURE
006612  000000                          HALT
006614  013767  177776  172162          MOV     @#PS,   ANS2    ;GET CPU STATUS
006622  042767  177760  172154          BIC     #177760,ANS2    ;SAVE CONDITION CODES
006630  016767  177750  172144          MOV     ANR56,  ANS1    ;SAVE THE ANSWER
006636  170200                          STFPS   FPS             ;STORE FLOATING POINT STATUS
006640  170367  172156                  STST    FEC             ;STORE EXCEPTION CODES
006644  022700  147605                  CMP     #147605,FPS     ;CHECK FLOATING POINT STATUS
006650  001401                          BEQ     .+4             ;BRANCH IF OK
006652  104000                          HLT                     ;FPS NOT EQUAL TO 147605

006654  022767  000006  172140          CMP     #6.,    FEC     ;CHECK FLOATING EXCEPTION CODE
```

```
006662  001401                          BEQ     .+4             ;BRANCH IF OK
006664  104000                          HLT                     ;FEC NOT EQUAL TO 6

006666  022767  006602  172130          CMP     #FPI56, FEA     ;CHECK FLOATING EXCEPTION ADDRESS
006674  001401                          BEQ     .+4             ;BRANCH IF OK
006676  104000                          HLT                     ;FEA NOT EQUAL TO FPI56

006700  022767  000000  172074          CMP     #000000,ANS1    ;DID 000000 GET STORED?
006706  001401                          BEQ     .+4             ;BRANCH IF OK
006710  104001                          HLT+1                   ;ANS1 NOT EQUAL TO 000000

006712  022767  000005  172064          CMP     #5,     ANS2    ;CONDITION CODES = 5?
006720  001401                          BEQ     .+4             ;BRANCH IF OK
006722  104002                          HLT+2                   ;WRONG CONDITION CODES!


                        ;*****************************************************************
                        ;TEST 57:       TEST STCDI (STORE-CONVERT DOUBLE TO INTEGER,
                        ;       STORE   100000,000000,000000,000000 --> 000000
                        ;       FPS = 047604,   AC = AC1,       DST = M6-R7
                        ;*****************************************************************

006724  104400                          SCOPE
006726  000404                          BR      TST57

006730  100000  000000  000000  DAT57:  100000.000000,000000,000000
006736  000000

006740  170127  040200          TST57:  LDFPS   #040200         ;LOAD FLOATING POINT STATUS
006744  172567  177760                  LDD     DAT57,  AC1     ;LOAD 100000,000000,000000,000000 INTO AC1
006750  170127  047600                  LDFPS   #047600         ;LOAD FLOATING POINT STATUS
006754  175567  172022                  STCDI   AC1,    ANS1    ;STORE-CONVERT AC1 IN ANS1
006760  013767  177776  172016          MOV     @#PS,   ANS2    ;GET CPU STATUS
006766  042767  177760  172010          BIC     #177760,ANS2    ;SAVE CONDITION CODES
006774  170200                          STFPS   FPS             ;STORE FLOATING POINT STATUS
006776  022700  047604                  CMP     #047604,FPS     ;CHECK FLOATING POINT STATUS
007002  001401                          BEQ     .+4             ;BRANCH IF OK
007004  104000                          HLT                     ;FPS NOT EQUAL TO 047604

007006  022767  000000  171766          CMP     #000000,ANS1    ;DID 000000 GET STORED?
007014  001401                          BEQ     .+4             ;BRANCH IF OK
007016  104001                          HLT+1                   ;ANS1 NOT EQUAL TO 000000

007020  022767  000004  171756          CMP     #4,     ANS2    ;CONDITION CODES = 4?
007026  001401                          BEQ     .+4             ;BRANCH IF OK
007030  104002                          HLT+2                   ;WRONG CONDITION CODES!


                        ;*****************************************************************
                        ;TEST 60:       TEST STCDI (STORE-CONVERT DOUBLE TO INTEGER)
                        ;       STORE   177777,177777,177777,177777 --> 000000
                        ;       FPS = 147605,   AC = AC0,       DST = M6-R7
                        ;       FEC = 6,        FEA = FPI60
                        ;*****************************************************************

007032  104400                          SCOPE
```

```
007034  000404                          BR      TST60

007036  177777  177777  177777  DAT60:  177777,177777,177777,177777
007044  177777

007046  170127  047600          TST60:  LDFPS   #047600         ;LOAD FLOATING POINT STATUS
007052  172467  177760                  LDD     DAT60,  AC0     ;LOAD 177777,177777,177777,177777 INTO AC0
007056  175467  171720          FPI60:  STCDI   AC0,    ANS1    ;STORE-CONVERT AC0 IN ANS1
007062  013767  177776  171714          MOV     @#PS,   ANS2    ;GET CPU STATUS
007070  042767  177760  171706          BIC     #177760,ANS2    ;SAVE CONDITION CODES
007076  170200                          STFPS   FPS             ;STORE FLOATING POINT STATUS
007100  170367  171716                  STST    FEC             ;STORE EXCEPTION CODES
007104  022700  147605                  CMP     #147605,FPS     ;CHECK FLOATING POINT STATUS
007110  001401                          BEQ     .+4             ;BRANCH IF OK
007112  104000                          HLT                     ;FPS NOT EQUAL TO 147605

007114  022767  000006  171700          CMP     #6,     FEC     ;CHECK FLOATING EXCEPTION CODE
007122  001401                          BEQ     .+4             ;BRANCH IF OK
007124  104000                          HLT                     ;FEC NOT EQUAL TO 6

007126  022767  007056  171670          CMP     #FPI60, FEA     ;CHECK FLOATING EXCEPTION ADDRESS
007134  001401                          BEQ     .+4             ;BRANCH IF OK
007136  104000                          HLT                     ;FEA NOT EQUAL TO FPI60

007140  022767  000000  171634          CMP     #000000,ANS1    ;DID 000000 GET STORED?
007146  001401                          BEQ     .+4             ;BRANCH IF OK
007150  104001                          HLT+1                   ;ANS1 NOT EQUAL TO 000000

007152  022767  000005  171624          CMP     #5,     ANS2    ;CONDITION CODES = 5?
007160  001401                          BEQ     .+4             ;BRANCH IF OK
007162  104002                          HLT+2                   ;WRONG CONDITION CODES!

                        ;*****************************************************************
                        ;TEST 61:       TEST STCDI (STORE-CONVERT DOUBLE TO INTEGER)
                        ;       STORE   000200,000000,000000,000000 --> 000000
                        ;       FPS = 047604,   AC = AC3,       DST = M6-R7
                        ;*****************************************************************

007164  104400                          SCOPE
007166  000404                          BR      TST61

007170  000200  000000  000000  DAT61:  000200,000000,000000,000000
007176  000000

007200  170127  047600          TST61:  LDFPS   #047600         ;LOAD FLOATING POINT STATUS
007204  172767  177760                  LDD     DAT61,  AC3     ;LOAD 000200,000000,000000,000000 INTO AC3
007210  175767  171566          FPI61:  STCDI   AC3,    ANS1    ;STORE-CONVERT AC3 IN ANS1
007214  013767  177776  171562          MOV     @#PS,   ANS2    ;GET CPU STATUS
007222  042767  177760  171554          BIC     #177760,ANS2    ;SAVE CONDITION CODES
007230  170200                          STFPS   FPS             ;STORE FLOATING POINT STATUS
007232  022700  047604                  CMP     #047604,FPS     ;CHECK FLOATING POINT STATUS
007236  001401                          BEQ     .+4             ;BRANCH IF OK
007240  104000                          HLT                     ;FPS NOT EQUAL TO 047604

007242  022767  000000  171532          CMP     #000000,ANS1    ;DID 000000 GET STORED?
```

```
007250  001401                          BEQ     .+4             ;BRANCH IF OK
007252  104001                          HLT+1                   ;ANS1 NOT EQUAL TO 000000

007254  022767  000004  171522          CMP     #4,     ANS2    ;CONDITION CODES = 4?
007262  001401                          BEQ     .+4             ;BRANCH IF OK
007264  104002                          HLT+2                   ;WRONG CONDITION CODES!


                        ;*****************************************************************
                        ;TEST 62:       TEST STCDI (STORE-CONVERT DOUBLE TO INTEGER)
                        ;       STORE   100177,177777,177777,177777 --> 000000
                        ;       FPS = 047604,   AC = AC2,       DST = M6-R7
                        ;*****************************************************************

007266  104400                          SCOPE
007270  000404                          BR      TST62

007272  100177  177777  177777  DAT62:  100177,177777,177777,177777
007300  177777

007302  170127  040200          TST62:  LDFPS   #040200         ;LOAD FLOATING POINT STATUS
007306  172667  177760                  LDD     DAT62,  AC2     ;LOAD 100177,177777,177777,177777 INTO AC2
007312  170127  047600                  LDFPS   #047600         ;LOAD FLOATING POINT STATUS
007316  175667  171460                  STCDI   AC2,    ANS1    ;STORE-CONVERT AC2 IN ANS1
007322  013767  177776  171454          MOV     @#PS,   ANS2    ;GET CPU STATUS
007330  042767  177760  171446          BIC     #177760,ANS2    ;SAVE CONDITION CODES
007336  170200                          STFPS   FPS             ;STORE FLOATING POINT STATUS
007340  022700  047604                  CMP     #047604,FPS     ;CHECK FLOATING POINT STATUS
007344  001401                          BEQ     .+4             ;BRANCH IF OK
007346  104000                          HLT                     ;FPS NOT EQUAL TO 047604

007350  022767  000000  171424          CMP     #000000,ANS1    ;DID 000000 GET STORED?
007356  001401                          BEQ     .+4             ;BRANCH IF OK
007360  104001                          HLT+1                   ;ANS1 NOT EQUAL TO 000000

007362  022767  000004  171414          CMP     #4,     ANS2    ;CONDITION CODES = 4?
007370  001401                          BEQ     .+4             ;BRANCH IF OK
007372  104002                          HLT+2                   ;WRONG CONDITION CODES!


                        ;*****************************************************************
                        ;TEST 63:       TEST LDCLF (LOAD-CONVERT LONG TO FLOATING)
                        ;       LOAD    000000,000000 --> 000000,000000
                        ;       FPS = 047504,   SRC = M6-R7,    AC = AC0
                        ;*****************************************************************

007374  104400                          SCOPE
007376  000402                          BR      TST63

007400  000000  000000          DAT63:  000000,000000

007404  170127  047500          TST63:  LDFPS   #047500         ;LOAD FLOATING POINT STATUS
007410  177067  177764                  LDCLF   DAT63,  AC0     ;LOAD-CONVERT 000000,000000 INTO AC0
007414  170200                          STFPS   FPS             ;STORE FLOATING POINT STATUS
007416  022700  047504                  CMP     #047504,FPS     ;CHECK FLOATING POINT STATUS
007422  001401                          BEQ     .+4             ;BRANCH IF OK
```

```
007424  104000                          HLT                     ;FPS NOT EQUAL TO 047504

007426  174067  171350                  STF     AC0,    ANS1    ;STORE AC0 IN ANS1, ANS2
007432  022767  000000  171342          CMP     #000000,ANS1    ;DID 000000 GET STORED?
007440  001401                          BEQ     .+4             ;BRANCH IF OK
007442  104002                          HLT+2                   ;ANS1 NOT EQUAL TO 000000

007444  022767  000000  171332          CMP     #000000,ANS2    ;DID 000000 GET STORED?
007452  001401                          BEQ     .+4             ;BRANCH IF OK
007454  104002                          HLT+2                   ;ANS2 NOT EQUAL TO 000000


                        ;*****************************************************************
                        ;TEST 64:       TEST LDCLF (LOAD-CONVERT LONG TO FLOATING
                        ;       LOAD    125252,125252 --> 147652,125253
                        ;       FPS = 047510,   SRC = M6-R7,    AC = AC3
                        ;*****************************************************************

007456  104400                          SCOPE
007460  000402                          BR      TST64

007462  125252  125253          DAT64:  125252,125252

007466  170127  047500          TST64:  LDFPS   #047500         ;LOAD FLOATING POINT STATUS
007472  177367  177764                  LDCLF   DAT64,  AC3     ;LOAD-CONVERT 125252,125252 INTO AC3
007476  170200                          STFPS   FPS             ;STORE FLOATING POINT STATUS
007500  022700  047510                  CMP     #047510,FPS     ;CHECK FLOATING POINT STATUS
007504  001401                          BEQ     .+4             ;BRANCH IF OK
007506  104000                          HLT                     ;FPS NOT EQUAL TO 047510

007510  174367  171266                  STF     AC3,    ANS1    ;STORE AC3 IN ANS1, ANS2
007514  022767  147652  171260          CMP     #147652,ANS1    ;DID 147652 GET STORED?
007522  001401                          BEQ     .+4             ;BRANCH IF OK
007524  104002                          HLT+2                   ;ANS1 NOT EQUAL TO 147652

007526  022767  125253  171250          CMP     #125253,ANS2    ;DID 125253 GET STORED?
007534  001401                          BEQ     .+4             ;BRANCH IF OK
007536  104002                          HLT+2                   ;ANS2 NOT EQUAL TO 125253


                        ;*****************************************************************
                        ;TEST 65:       TEST LDCLF (LOAD-CONVERT LONG TO FLOATING,
                        ;       LOAD    052525,052525 --> 047652,125253
                        ;       FPS = 047500,   SRC = M6-R7,    AC = AC1
                        ;*****************************************************************

007540  104400                          SCOPE
007542  000402                          BR      TST65

007544  052525  052525          DAT65:  052525,052525

007550  170127  047500          TST65:  LDFPS   #047500         ;LOAD FLOATING POINT STATUS
007554  177167  177764                  LDCLF   DAT65,  AC1     ;LOAD-CONVERT 052525,052525 INTO AC1
007560  170200                          STFPS   FPS             ;STORE FLOATING POINT STATUS
007562  022700  047500                  CMP     #047500,FPS     ;CHECK FLOATING POINT STATUS
007570  001401                          BEQ     .+4             ;BRANCH IF OK
```

```
007570  104000                          HLT                     ;FPS NOT EQUAL TO 047500

007572  174167  171204                  STF     AC1,    ANS1    ;STORE AC1 IN ANS1, ANS2
007576  022767  047652  171176          CMP     #047652,ANS1    ;DID 047652 GET STORED?
007604  001401                          BEQ     .+4             ;BRANCH IF OK
007606  104002                          HLT+2                   ;ANS1 NOT EQUAL TO 047652

007610  022767  125253  171166          CMP     #125253,ANS2    ;DID 125253 GET STORED?
007616  001401                          BEQ     .+4             ;BRANCH IF OK
007620  104002                          HLT+2                   ;ANS2 NOT EQUAL TO 125253


                        ;*****************************************************************
                        ;TEST 66:       TEST LDCLF (LOAD-CONVERT LONG TO FLOATING)
                        ;       LOAD    000000,000001 --> 040200,000000
                        ;       FPS = 047500,   SRC = M6-R7,    AC = AC3
                        ;*****************************************************************

007622  104400                          SCOPE
007624  000402                          BR      TST66

007626  000000  000001          DAT66:  000000,000001

007632  170127  047500          TST66:  LDFPS   #047500         ;LOAD FLOATING POINT STATUS
007636  177367  177764                  LDCLF   DAT66,  AC3     ;LOAD-CONVERT 000000,000001 INTO AC3
007642  170200                          STFPS   FPS             ;STORE FLOATING POINT STATUS
007644  022700  047500                  CMP     #047500,FPS     ;CHECK FLOATING POINT STATUS
007650  001401                          BEQ     .+4             ;BRANCH IF OK
007652  104000                          HLT                     ;FPS NOT EQUAL TO 047500

007654  174367  171122                  STF     AC3,    ANS1    ;STORE AC3 IN ANS1, ANS2
007660  022767  040200  171114          CMP     #040200,ANS1    ;DID 040200 GET STORED?
007666  001401                          BEQ     .+4             ;BRANCH IF OK
007670  104002                          HLT+2                   ;ANS1 NOT EQUAL TO 040200

007672  022767  000000  171104          CMP     #000000,ANS2    ;DID 000000 GET STORED?
007700  001401                          BEQ     .+4             ;BRANCH IF OK
007702  104002                          HLT+2                   ;ANS2 NOT EQUAL TO 000000


                        ;*****************************************************************
                        ;TEST 67:       TEST LDCLF (LOAD-CONVERT LONG TO FLOATING)
                        ;       LOAD    177777 --> 144200,000000
                        ;       FPS = 047510,   SRC = M2-R7,    AC = AC2
                        ;*****************************************************************

007704  104400                          SCOPE
007706  170127  047500          TST67:  LDFPS   #047500         ;LOAD FLOATING POINT STATUS
007712  177227  177777                  LDCLF   #177777,AC2     ;LOAD-CONVERT 177777 INTO AC2
007716  170200                          STFPS   FPS             ;STORE FLOATING POINT STATUS
007720  022700  047510                  CMP     #047510,FPS     ;CHECK FLOATING POINT STATUS
007724  001401                          BEQ     .+4             ;BRANCH IF OK
007726  104000                          HLT                     ;FPS NOT EQUAL TO 047510

007730  174267  171046                  STF     AC2,    ANS1    ;STORE AC2 IN ANS1, ANS2
007734  022767  144200  171040          CMP     #144200,ANS1    ;DID 144200 GET STORED?
```

```
                                    BEQ     .+4                 ;BRANCH IF OK
                                    HLT+2                       ;ANS1 NOT EQUAL TO 144200

                                    CMP     #000000,ANS2        ;DID 000000 GET STORED?
                                    BEQ     .+4                 ;BRANCH IF OK
                                    HLT+2                       ;ANS2 NOT EQUAL TO 000000

                        ;***********************************************************************
                        ;TEST 70:       TEST LDCLF (LOAD-CONVERT LONG TO FLOATING)
                        ;       LOAD    100000 --> 150000,000000
                        ;       FPS = 047510,   SRC = M0-R4,    AC = AC0
                        ;***********************************************************************

                                    SCOPE
007760  104400          TST70:    
007762  170127  047500            LDFPS   #047500             ;LOAD FLOATING POINT STATUS
007766  012704  100000            MOV     #100000,R4          ;"LOAD" 100000 INTO R4
007772  177004                    LDCLF   R4,     AC0         ;LOAD-CONVERT 100000 INTO AC0
007774  170200                    STFPS   FPS                 ;STORE FLOATING POINT STATUS
007776  022700  047510            CMP     #047510,FPS         ;CHECK FLOATING POINT STATUS
010002  001401                    BEQ     .+4                 ;BRANCH IF OK
010004  104000                    HLT                         ;FPS NOT EQUAL TO 047510

010006  174067  170770            STF     AC0,    ANS1        ;STORE AC0 IN ANS1, ANS2
010012  022767  150000  170762    CMP     #150000,ANS1        ;DID 150000 GET STORED?
010020  001401                    BEQ     .+4                 ;BRANCH IF OK
010022  104002                    HLT+2                       ;ANS1 NOT EQUAL TO 150000

010024  022767  000000  170752    CMP     #000000,ANS2        ;DID 000000 GET STORED?
010032  001401                    BEQ     .+4                 ;BRANCH IF OK
010034  104002                    HLT+2                       ;ANS2 NOT EQUAL TO 000000

                        ;***********************************************************************
                        ;TEST 71:       TEST LDCLF (LOAD-CONVERT LONG TO FLOATING)
                        ;       LOAD    077777,177777 --> 050000,000000
                        ;       FPS = 047500,   SRC = M6-R7,    AC = AC2
                        ;***********************************************************************

                                    SCOPE
                                    BR      TST71

010042  077777  177777    DAT71:  077777,177777

010046  170127  047500    TST71:  LDFPS   #047500             ;LOAD FLOATING POINT STATUS
010052  177267  177764            LDCLF   DAT71,  AC2         ;LOAD-CONVERT 077777,177777 INTO AC2
010056  170200                    STFPS   FPS                 ;STORE FLOATING POINT STATUS
010060  022700  047500            CMP     #047500,FPS         ;CHECK FLOATING POINT STATUS
010064  001401                    BEQ     .+4                 ;BRANCH IF OK
010066  104000                    HLT                         ;FPS NOT EQUAL TO 047500

010070  174267  170706            STF     AC2,    ANS1        ;STORE AC2 IN ANS1, ANS2
010074  022767  050000  170700    CMP     #050000,ANS1        ;DID 050000 GET STORED?
010102  001401                    BEQ     .+4                 ;BRANCH IF OK
010104  104002                    HLT+2                       ;ANS1 NOT EQUAL TO 050000
```

```
010136  022767  000000  170670          CMP     #000000,ANS2    ;DID 000000 GET STORED?
010144  001401                          BEQ     .+4             ;BRANCH IF OK
010146  104002                          HLT+2                   ;ANS2 NOT EQUAL TO 000000


                        ;*****************************************************************
                        ;TEST 72:       TEST LDCLF (LOAD-CONVERT LONG TO FLOATING)
                        ;       LOAD    000125,000125 --> 045652,000252
                        ;       FPS = 047500,   SRC = M6-P7,    AC = AC0
                        ;*****************************************************************

010150  104422                          SCOPE
010152  000402                          BR      .+TST72

010154  000125  000125          DAT72:  000125,000125

010160  170127  047500          TST72:  LDFPS   #047500         ;LOAD FLOATING POINT STATUS
010164  177067  177764                  LDCLF   DAT72,  AC0     ;LOAD-CONVERT 000125,000125 INTO AC0
010170  170200                          STFPS   FPS             ;STORE FLOATING POINT STATUS
010172  022700  047500                  CMP     #047500,FPS     ;CHECK FLOATING POINT STATUS
010176  001401                          BEQ     .+4             ;BRANCH IF OK
010200  104000                          HLT                     ;FPS NOT EQUAL TO 047500

010202  174067  170624                  STF     AC0,    ANS1    ;STORE AC0 IN ANS1, ANS2
010206  022767  045652  170616          CMP     #045652,ANS1    ;DID 045652 GET STORED?
010214  001401                          BEQ     .+4             ;BRANCH IF OK
010216  104002                          HLT+2                   ;ANS1 NOT EQUAL TO 045652

010220  022767  000252  170606          CMP     #000252,ANS2    ;DID 000252 GET STORED?
010226  001401                          BEQ     .+4             ;BRANCH IF OK
010230  104002                          HLT+2                   ;ANS2 NOT EQUAL TO 000252


                        ;*****************************************************************
                        ;TEST 73:       TEST LDCLF (LOAD-CONVERT LONG TO FLOATING)
                        ;       LOAD    070707,070707 --> 047743,107344
                        ;       FPS = 047500,   SRC = M6-R7,    AC = AC2
                        ;*****************************************************************

010232  104422                          SCOPE
010234  000402                          BR      .+TST73

010236  070707  070707          DAT73:  070707,070707

010242  170127  047500          TST73:  LDFPS   #047500         ;LOAD FLOATING POINT STATUS
010246  177267  177764                  LDCLF   DAT73,  AC2     ;LOAD-CONVERT 070707,070707 INTO AC2
010252  170200                          STFPS   FPS             ;STORE FLOATING POINT STATUS
010254  022700  047500                  CMP     #047500,FPS     ;CHECK FLOATING POINT STATUS
010260  001401                          BEQ     .+4             ;BRANCH IF OK
010262  104000                          HLT                     ;FPS NOT EQUAL TO 047500

010264  174267  170542                  STF     AC2,    ANS1    ;STORE AC2 IN ANS1, ANS2
010270  022767  047743  170534          CMP     #047743,ANS1    ;DID 047743 GET STORED?
010276  001401                          BEQ     .+4             ;BRANCH IF OK
010300  104002                          HLT+2                   ;ANS1 NOT EQUAL TO 047743
```

```
010252  022767  107344  170524          CMP     #107344,ANS2    ;DID 107344 GET STORED?
010260  001401                          BEQ     .+4             ;BRANCH IF OK
010262  104002                          HLT+2                   ;ANS2 NOT EQUAL TO 107344


                                ;*****************************************************************
                                ;TEST 74:       TEST LDCLF (LOAD-CONVERT LONG TO FLOATING)
                                ;       LOAD    107070,107070 --> 147743,107344
                                ;       FPS = 047510,   SRC = M6-R7,    AC = AC0
                                ;*****************************************************************

010264  104400                          SCOPE
010266  000402                          BR      TST74

010270  107070  107070          DAT74:  107070,107070

010274  170127  047500          TST74:  LDFPS   #047500         ;LOAD FLOATING POINT STATUS
010300  177067  177764                  LDCLF   DAT74,  AC0     ;LOAD-CONVERT 107070,107070 INTO AC0
010304  170200                          STFPS   FPS             ;STORE FLOATING POINT STATUS
010306  022700  047510                  CMP     #047510,FPS     ;CHECK FLOATING POINT STATUS
010312  001401                          BEQ     .+4             ;BRANCH IF OK
010314  104000                          HLT                     ;FPS NOT EQUAL TO 047510

010316  174067  170460                  STF     AC0,    ANS1    ;STORE AC0 IN ANS1, ANS2
010322  022757  147743  170452          CMP     #147743,ANS1    ;DID 147743 GET STORED?
010330  001401                          BEQ     .+4             ;BRANCH IF OK
010332  104002                          HLT+2                   ;ANS1 NOT EQUAL TO 147743
010334  022767  107344  170442          CMP     #107344,ANS2    ;DID 107344 GET STORED?
010342  001401                          BEQ     .+4             ;BRANCH IF OK
010344  104002                          HLT+2                   ;ANS2 NOT EQUAL TO 107344


                                ;*****************************************************************
                                ;TEST 75:       TEST STCFL (STORE-CONVERT FLOATING TO LONG)
                                ;       STORE   000000,000000 --> 000000,000000
                                ;       FPS = 047504,   AC = AC2,       DST = M6-R7
                                ;*****************************************************************

010346  104400                          SCOPE
010350  000402                          BR      TST75

010352  000000  000000          DAT75:  000000,000000

010356  170127  047500          TST75:  LDFPS   #047500         ;LOAD FLOATING POINT STATUS
010362  172667  177764                  LDF     DAT75,  AC2     ;LOAD 000000,000000 INTO AC2
010366  175667  170410          FPI75:  STCFL   AC2,    ANS1    ;STORE-CONVERT AC2 IN ANS1, ANS2
010372  013767  177776  170406          MOV     @#PS,   ANS3    ;GET CPU STATUS
010400  042767  177760  170400          BIC     #177760,ANS3    ;SAVE CONDITION CODES
010406  170200                          STFPS   FPS             ;STORE FLOATING POINT STATUS
010410  022700  047504                  CMP     #047504,FPS     ;CHECK FLOATING POINT STATUS
010414  001401                          BEQ     .+4             ;BRANCH IF OK
010416  104000                          HLT                     ;FPS NOT EQUAL TO 047504

010420  022767  000000  170354          CMP     #000000,ANS1    ;DID 000000 GET STORED?
010426  001401                          BEQ     .+4             ;BRANCH IF OK
```

```
010430  104002                          HLT+2                   ;ANS1 NOT EQUAL TO 000000

010432  022767  000000  170344          CMP     #000000,ANS2    ;DID 000000 GET STORED?
010440  001401                          BEQ     .+4             ;BRANCH IF OK
010442  104002                          HLT+2                   ;ANS2 NOT EQUAL TO 000000

010444  022767  000004  170334          CMP     #4,     ANS3    ;CONDITION CODES = 4?
010452  001401                          BEQ     .+4             ;BRANCH IF OK
010454  104003                          HLT+3                   ;WRONG CONDITION CODES!


                        ;*****************************************************************
                        ;TEST 76:       TEST STCFL (STORE-CONVERT FLOATING TO LONG.
                        ;       STORE   041252.125252 --> 000000,000025
                        ;       FPS = 047500,   AC = AC1,       OST = M6-R7
                        ;*****************************************************************

010456  104400                          SCOPE
010460  000402                          BR      TST76

010462  041252  125252          DAT76:  041252.125252

010466  170127  047500          TST76:  LDFPS   #047500         ;LOAD FLOATING POINT STATUS
010472  172567  177764                  LDF     DAT76,  AC1     ;LOAD 041252,125252 INTO AC1
010476  175567  170300          FPI76:  STCFL   AC1,    ANS1    ;STORE-CONVERT AC1 IN ANS1, ANS2
010502  013767  177776  170276          MOV     @#PS,   ANS3    ;GET CPU STATUS
010510  042767  177760  170270          BIC     #177760,ANS3    ;SAVE CONDITION CODES
010516  170200                          STFPS   FPS             ;STORE FLOATING POINT STATUS
010520  022700  047500                  CMP     #047500,FPS     ;CHECK FLOATING POINT STATUS
010524  001401                          BEQ     .+4             ;BRANCH IF OK
010526  104000                          HLT                     ;FPS NOT EQUAL TO 047500

010530  022767  000000  170244          CMP     #000000,ANS1    ;DID 000000 GET STORED?
010536  001401                          BEQ     .+4             ;BRANCH IF OK
010540  104002                          HLT+2                   ;ANS1 NOT EQUAL TO 000000

010542  022767  000025  170234          CMP     #000025,ANS2    ;DID 000025 GET STORED?
010550  001401                          BEQ     .+4             ;BRANCH IF OK
010552  104002                          HLT+2                   ;ANS2 NOT EQUAL TO 000025

010554  022767  000000  170224          CMP     #0,     ANS3    ;CONDITION CODES = 0?
010562  001401                          BEQ     .+4             ;BRANCH IF OK
010564  104003                          HLT+3                   ;WRONG CONDITION CODES!


                        ;*****************************************************************
                        ;TEST 77:       TEST STCFL (STORE-CONVERT FLOATING TO LONG
                        ;       STORE   141252.125252 --> 177777,177753
                        ;       FPS = 047510,   AC = AC3,       OST = M6-R7
                        ;*****************************************************************

010566  104400                          SCOPE
010570  000402                          BR      TST77

010572  141252  125252          DAT77:  141252.125252
```

```
010576  170127  047500          TST77:  LDFPS   #047500         ;LOAD FLOATING POINT STATUS
010602  172767  177764                  LDF     DAT77, AC3      ;LOAD 141252,125252 INTO AC3
010606  175767  170170          FPI77:  STCFL   AC3,    ANS1    ;STORE-CONVERT AC3 IN ANS1, ANS2
010612  013767  177776  170166          MOV     @#PS,   ANS3    ;GET CPU STATUS
010620  042767  177760  170160          BIC     #177760,ANS3    ;SAVE CONDITION CODES
010626  170200                          STFPS   FPS             ;STORE FLOATING POINT STATUS
010630  022700  047510                  CMP     #047510,FPS     ;CHECK FLOATING POINT STATUS
010634  001401                          BEQ     .+4             ;BRANCH IF OK
010636  104000                          HLT                     ;FPS NOT EQUAL TO 047510

010640  022767  177777  170134          CMP     #177777,ANS1    ;DID 177777 GET STORED?
010646  001401                          BEQ     .+4             ;BRANCH IF OK
010650  104002                          HLT+2                   ;ANS1 NOT EQUAL TO 177777

010652  022767  177753  170124          CMP     #177753,ANS2    ;DID 177753 GET STORED?
010660  001401                          BEQ     .+4             ;BRANCH IF OK
010662  104002                          HLT+2                   ;ANS2 NOT EQUAL TO 177753

010664  022767  000010  170114          CMP     #10.    ANS3    ;CONDITION CODES = 10?
010672  001401                          BEQ     .+4             ;BRANCH IF OK
010674  104003                          HLT+3                   ;WRONG CONDITION CODES!


                                ;*****************************************************************
                                ;TST 100:       TEST STCFL (STORE-CONVERT FLOATING TO LONG.
                                ;       STORE   040177,177777 --> 000000,000000
                                ;       FPS = 047504,   AC = AC1,       DST = M6-R?
                                ;*****************************************************************

010676  104400                          SCOPE
010700  000402                          BR      TST100

010702  040177  177777          DAT100: 040177,177777

010706  170127  047500          TST100: LDFPS   #047500         ;LOAD FLOATING POINT STATUS
010712  172567  177764                  LDF     DAT100, AC1     ;LOAD 040177,177777 INTO AC1
010716  175567  170060          FPI100: STCFL   AC1,    ANS1    ;STORE-CONVERT AC1 IN ANS1, ANS2
010722  013767  177776  170056          MOV     @#PS,   ANS3    ;GET CPU STATUS
010730  042767  177760  170050          BIC     #177760,ANS3    ;SAVE CONDITION CODES
010736  170200                          STFPS   FPS             ;STORE FLOATING POINT STATUS
010740  022700  047504                  CMP     #047504,FPS     ;CHECK FLOATING POINT STATUS
010744  001401                          BEQ     .+4             ;BRANCH IF OK
010746  104000                          HLT                     ;FPS NOT EQUAL TO 047504

010750  022767  000000  170024          CMP     #000000,ANS1    ;DID 000000 GET STORED?
010756  001401                          BEQ     .+4             ;BRANCH IF OK
010760  104002                          HLT+2                   ;ANS1 NOT EQUAL TO 000000

010762  022767  000000  170014          CMP     #000000,ANS2    ;DID 000000 GET STORED?
010770  001401                          BEQ     .+4             ;BRANCH IF OK
010772  104002                          HLT+2                   ;ANS2 NOT EQUAL TO 000000

010774  022767  000004  170004          CMP     #4.     ANS3    ;CONDITION CODES = 4?
011002  001401                          BEQ     .+4             ;BRANCH IF OK
011004  104003                          HLT+3                   ;WRONG CONDITION CODES!
```

```
                        ;*****************************************************************
                        ;TEST 101:      TEST STCFL (STORE-CONVERT FLOATING TO LONG)
                        ;       STORE   140177,177777 --> 000000,000000
                        ;       FPS = 047504,   AC = AC2,       DST = M6-R7
                        ;*****************************************************************

011006  104400                  SCOPE
011010  000402                  BR      TST101

011012  140177  177777    DAT101: 140177,177777

011016  170127  047500    TST101: LDFPS   #047500         ;LOAD FLOATING POINT STATUS
011022  172667  177764            LDF     DAT101, AC2     ;LOAD 140177,177777 INTO AC2
011026  175667  167750    FPI101: STCFL   AC2,    ANS1    ;STORE-CONVERT AC2 IN ANS1, ANS2
011032  013767  177776  167746        MOV     @#PS,   ANS3    ;GET CPU STATUS
011040  042767  177760  167740        BIC     #177760,ANS3    ;SAVE CONDITION CODES
011046  170200                  STFPS   FPS             ;STORE FLOATING POINT STATUS
011050  022700  047504            CMP     #047504,FPS     ;CHECK FLOATING POINT STATUS
011054  001401                  BEQ     .+4             ;BRANCH IF OK
011056  104000                  HLT                     ;FPS NOT EQUAL TO 047504

011060  022767  000000  167714        CMP     #000000,ANS1    ;DID 000000 GET STORED?
011066  001401                  BEQ     .+4             ;BRANCH IF OK
011070  104002                  HLT+2                   ;ANS1 NOT EQUAL TO 000000

011072  022767  000000  167704        CMP     #000000,ANS2    ;DID 000000 GET STORED?
011100  001401                  BEQ     .+4             ;BRANCH IF OK
011102  104002                  HLT+2                   ;ANS2 NOT EQUAL TO 000000

011104  022767  000004  167674        CMP     #4,     ANS3    ;CONDITION CODES = 4?
011112  001401                  BEQ     .+4             ;BRANCH IF OK
011114  104003                  HLT+3                   ;WRONG CONDITION CODES!


                        ;*****************************************************************
                        ;TEST 102:      TEST STCFL (STORE-CONVERT FLOATING TO LONG)
                        ;       STORE   040200,000001 --> 000000,000001
                        ;       FPS = 047500,   AC = AC0,       DST = M6-R7
                        ;*****************************************************************

011116  104400                  SCOPE
011120  000402                  BR      TST102

011122  040200  000001    DAT102: 040200,000001

011126  170127  047500    TST102: LDFPS   #047500         ;LOAD FLOATING POINT STATUS
011132  172467  177764            LDF     DAT102, AC0     ;LOAD 040200,000001 INTO AC0
011136  175467  167640    FPI102: STCFL   AC0,    ANS1    ;STORE-CONVERT AC0 IN ANS1, ANS2
011142  013767  177776  167636        MOV     @#PS,   ANS3    ;GET CPU STATUS
011150  042767  177760  167630        BIC     #177760,ANS3    ;SAVE CONDITION CODES
011156  170200                  STFPS   FPS             ;STORE FLOATING POINT STATUS
011160  022700  047500            CMP     #047500,FPS     ;CHECK FLOATING POINT STATUS
011164  001401                  BEQ     .+4             ;BRANCH IF OK
011166  104000                  HLT                     ;FPS NOT EQUAL TO 047500
```

```
011170  022767  000000  167604          CMP     #000000,ANS1    ;DID 000000 GET STORED?
011176  001401                          BEQ     .+4             ;BRANCH IF OK
011200  104002                          HLT+2                   ;ANS1 NOT EQUAL TO 000000

011202  022767  000001  167574          CMP     #000001,ANS2    ;DID 000001 GET STORED?
011210  001401                          BEQ     .+4              BRANCH IF OK
011212  104002                          HLT+2                   ;ANS2 NOT EQUAL TO 000001

011214  022767  000000  167564          CMP     #0,     ANS3    ;CONDITION CODES = 0?
011222  001401                          BEQ     .+4             ;BRANCH IF OK
011224  104003                          HLT+3                   ;WRONG CONDITION CODES!


                        ;*****************************************************************************
                        ;TEST 103:      TEST STCFL (STORE-CONVERT FLOATING TO LONG)
                        ;       STORE   140200,000001 --> 177777,177777
                        ;       FPS = 047510,   AC = AC2,       DST = M6-R7
                        ;*****************************************************************************

011226  104400                          SCOPE
011230  000402                          BR      TST103

011232  140200  000001  DAT103: 140200,000001

011236  170127  047500  TST103: LDFPS   #047500                 ;LOAD FLOATING POINT STATUS
011242  172667  177764          LDF     DAT103, AC2             ;LOAD 140200,000001 INTO AC2
011246  175667  167530  FPI103: STCFL   AC2,     ANS1           ;STORE-CONVERT AC2 IN ANS1, ANS2
011252  013767  177776  167526          MOV     @#PS,   ANS3    ;GET CPU STATUS
011260  042767  177760  167520          BIC     #177760,ANS3    ;SAVE CONDITION CODES
011266  170200                          STFPS   FPS             ;STORE FLOATING POINT STATUS
011270  022700  047510                  CMP     #047510,FPS     ;CHECK FLOATING POINT STATUS
011274  001401                          BEQ     .+4             ;BRANCH IF OK
011276  104000                          HLT                     ;FPS NOT EQUAL TO 047510

011300  022767  177777  167474          CMP     #177777,ANS1    ;DID 177777 GET STORED?
011306  001401                          BEQ     .+4             ;BRANCH IF OK
011310  104002                          HLT+2                   ;ANS1 NOT EQUAL TO 177777

011312  022767  177777  167464          CMP     #177777,ANS2    ;DID 177777 GET STORED?
011320  001401                          BEQ     .+4             ;BRANCH IF OK
011322  104002                          HLT+2                   ;ANS2 NOT EQUAL TO 177777

011324  022767  000010  167454          CMP     #10.    ANS3    ;CONDITION CODES = 10?
011332  001401                          BEQ     .+4             ;BRANCH IF OK
011334  104003                          HLT+3                   ;WRONG CONDITION CODES!


                        ;*****************************************************************************
                        ;TEST 104:      TEST STCFL (STORE-CONVERT FLOATING TO LONG)
                        ;       STORE   047777,177777 --> 077777
                        ;       FPS = 047500,   AC = AC0,       DST = M2-R7
                        ;*****************************************************************************

011336  104400                          SCOPE
011340  000402                          BR      TST104
```

```
011342  047777  177777          DAT104: 047777,177777

011346  170127  047500          TST104: LDFPS   #C47500          ;LOAD FLOATING POINT STATUS
011352  172467  177764                  LDF     DAT104, AC0      ;LOAD 047777,177777 INTO AC0
011356  175427                  ANR104: STCFL   AC0,    (PC)+    ;STORE-CONVERT AC0 IN +2
011360  000000                          0
011362  000402                          BR      .+6
011364  000000                          HALT                     ;PC FAILURE
011366  000000                          HALT
011370  013767  177776  167406        MOV     @#PS,   ANS2     ;GET CPU STATUS
011376  042767  177760  167400        BIC     #177760,ANS2     ;SAVE CONDITION CODES
011404  016767  177750  167370        MOV     ANR104, ANS1     ;SAVE THE ANSWER
011412  170200                        STFPS   FPS              ;STORE FLOATING POINT STATUS
011414  022700  047500                CMP     #047500,FPS      ;CHECK FLOATING POINT STATUS
011420  001401                        BEQ     .+4              ;BRANCH IF OK
011422  104000                        HLT                      ;FPS NOT EQUAL TO 047500

011424  022767  077777  167350        CMP     #077777,ANS1     ;DID 077777 GET STORED?
011432  001401                        BEQ     .+4              ;BRANCH IF OK
011434  104001                        HLT+1                    ;ANS1 NOT EQUAL TO 077777

011436  022767  000000  167340        CMP     #0,     ANS2     ;CONDITION CODES = 0?
011444  001401                        BEQ     .+4              ;BRANCH IF OK
011446  104002                        HLT+2                    ;WRONG CONDITION CODES!
```

```
;*****************************************************************
;TEST 105:      TEST STCFL (STORE-CONVERT FLOATING TO LONG)
;       STORE   050000,000000 --> 000000
;       FPS = 147505,   AC = AC2,       DST = MO-R5
;       FEC = 6,        FEA = FPI105
;*****************************************************************

011450  104400                        SCOPE
011452  000402                        BR      TST105

011454  050000  000000          DAT105: 050000,000000

011460  170127  047500          TST105: LDFPS   #047500          ;LOAD FLOATING POINT STATUS
011464  172667  177764                  LDF     DAT105, AC2      ;LOAD 050000,000000 INTO AC2
011470  175605                  FPI105: STCFL   AC2,    R5       ;STORE-CONVERT AC2 IN R5
011472  013767  177776  167304        MOV     @#PS,   ANS2     ;GET CPU STATUS
011500  042767  177760  167276        BIC     #177760,ANS2     ;SAVE CONDITION CODES
011506  010567  167270                MOV     R5,     ANS1     ;SAVE R5
011512  170200                        STFPS   FPS              ;STORE FLOATING POINT STATUS
011514  170367  167302                STST    FEC              ;STORE EXCEPTION CODES
011520  022700  147505                CMP     #147505,FPS      ;CHECK FLOATING POINT STATUS
011524  001401                        BEQ     .+4              ;BRANCH IF OK
011526  104000                        HLT                      ;FPS NOT EQUAL TO 147505

011530  022767  000006  167264        CMP     #6,     FEC      ;CHECK FLOATING EXCEPTION CODE
011536  001401                        BEQ     .+4              ;BRANCH IF OK
011540  104000                        HLT                      ;FEC NOT EQUAL TO 6

011542  022767  011470  167254        CMP     #FPI105, FEA     ;CHECK FLOATING EXCEPTION ADDRESS
011550  001401                        BEQ     .+4              ;BRANCH IF OK
```

```
011552  104000                          HLT                     ;FEA NOT EQUAL TO FPI105

011554  022767  000000  167220          CMP     #000000,ANS1    ;DID 000000 GET STORED?
011562  001401                          BEQ     .+4             ;BRANCH IF OK
011564  104001                          HLT+1                   ;ANS1 NOT EQUAL TO 000000

011566  022767  000005  167210          CMP     #5,     ANS2    ;CONDITION CODES = 5?
011574  001401                          BEQ     .+4             ;BRANCH IF OK
011576  104002                          HLT+2                   ;WRONG CONDITION CODES!


                        ;*****************************************************************************
                        ;TEST 106:      TEST STCFL (STORE-CONVERT FLOATING TO LONG)
                        ;       STORE   150000,000000 --> 100000,000000
                        ;       FPS = 047510,   AC = AC2,       DST = M6-R7
                        ;*****************************************************************************

011600  104400                          SCOPE
011602  000402                          BR      TST106

011604  150000  000000          DAT106: 150000,000000

011610  170127  047500          TST106: LDFPS   #047500         ;LOAD FLOATING POINT STATUS
011614  172667  177764                  LDF     DAT106, AC2     ;LOAD 150000,000000 INTO AC2
011620  175667  167156          FPI106: STCFL   AC2,    ANS1    ;STORE-CONVERT AC2 IN ANS1, ANS2
011624  013767  177776  167154          MOV     @#PS,   ANS3    ;GET CPU STATUS
011632  042767  177760  167146          BIC     #177760,ANS3    ;SAVE CONDITION CODES
011640  170200                          STFPS   FPS             ;STORE FLOATING POINT STATUS
011642  022700  047510                  CMP     #047510,FPS     ;CHECK FLOATING POINT STATUS
011646  001401                          BEQ     .+4             ;BRANCH IF OK
011650  104000                          HLT                     ;FPS NOT EQUAL TO 047510

011652  022767  100000  167122          CMP     #100000,ANS1    ;DID 100000 GET STORED?
011660  001401                          BEQ     .+4             ;BRANCH IF OK
011662  104002                          HLT+2                   ;ANS1 NOT EQUAL TO 100000

011664  022767  000000  167112          CMP     #000000,ANS2    ;DID 000000 GET STORED?
011672  001401                          BEQ     .+4             ;BRANCH IF OK
011674  104002                          HLT+2                   ;ANS2 NOT EQUAL TO 000000

011676  022767  000010  167102          CMP     #10.,   ANS3    ;CONDITION CODES = 10?
011704  001401                          BEQ     .+4             ;BRANCH IF OK
011706  104003                          HLT+3                   ;WRONG CONDITION CODES!


                        ;*****************************************************************************
                        ;TEST 107:      TEST STCFL (STORE-CONVERT FLOATING TO LONG)
                        ;       STORE   150000,000001 --> 000000,000000
                        ;       FPS = 147505,   AC = AC1,       DST = M6-R7
                        ;       FEC = 6,        FEA = FPI107
                        ;*****************************************************************************

011710  104400                          SCOPE
011712  000402                          BR      TST107

011714  150000  000001          DAT107: 150000,000001
```

```
011720  170127  047500          TST107: LDFPS   #047500          ;LOAD FLOATING POINT STATUS
011724  172567  177764                  LDF     DAT107, AC1     ;LOAD 150000,00000! INTO AC1
011730  175567  167046          FPI107: STCFL   AC1,    ANS1    ;STORE-CONVERT AC1 IN ANS1, ANS2
011734  013767  177776  167044          MOV     @#PS,   ANS3    ;GET CPU STATUS
011742  042767  177760  167036          BIC     #177760,ANS3    ;SAVE CONDITION CODES
011750  170200                          STFPS   FPS             ;STORE FLOATING POINT STATUS
011752  170367  167044                  STST    FEC             ;STORE EXCEPTION CODES
011756  022700  147505                  CMP     #147505,FPS     ;CHECK FLOATING POINT STATUS
011762  001401                          BEQ     .+4             ;BRANCH IF OK
011764  104000                          HLT                     ;FPS NOT EQUAL TO 147505

011766  022767  000006  167026          CMP     #6,     FEC     ;CHECK FLOATING EXCEPTION CODE
011774  001401                          BEQ     .+4             ;BRANCH IF OK
011776  104000                          HLT                     ;FEC NOT EQUAL TO 6

012000  022767  011730  167016          CMP     #FPI1C7, FEA    ;CHECK FLOATING EXCEPTION ADDRESS
012006  001401                          BEQ     .+4             ;BRANCH IF OK
012010  104000                          HLT                     ;FEA NOT EQUAL TO FPI107

012012  022767  000000  166762          CMP     #000000,ANS1    ;DID 000000 GET STORED?
012020  001401                          BEQ     .+4             ;BRANCH IF OK
012022  104002                          HLT+2                   ;ANS1 NOT EQUAL TO 000000

012024  022767  000000  166752          CMP     #000000,ANS2    ;DID 000000 GET STORED?
012032  001401                          BEQ     .+4             ;BRANCH IF OK
012034  104002                          HLT+2                   ;ANS2 NOT EQUAL TO 000000

012036  022767  000005  166742          CMP     #5,     ANS3    ;CONDITION CODES = 5?
012044  001401                          BEQ     .+4             ;BRANCH IF OK
012046  104003                          HLT+3                   ;WRONG CONDITION CODES!

                        ;*****************************************************************
                        ;TEST 110:      TEST STCFL (STORE-CONVERT FLOATING TO LONG)
                        ;       STORE   100000,000000 --> 000000,000000
                        ;       FPS = 047504,   AC = AC0,       DST = M6-R7
                        ;*****************************************************************

012050  104400                          SCOPE
012052  000402                          BR      TST110

012054  100000  000000  DAT110: 100000,000000

012060  170127  040000  TST110: LDFPS   #040000          ;LOAD FLOATING POINT STATUS
012064  172467  177764          LDF     DAT110, AC0     ;LOAD 100000,000000 INTO AC0
012070  170127  047500          LDFPS   #047500         ;LOAD FLOATING POINT STATUS
012074  175467  166702          STCFL   AC0,    ANS1    ;STORE-CONVERT AC0 IN ANS1, ANS2
012100  013767  177776  166700  MOV     @#PS,   ANS3    ;GET CPU STATUS
012106  042767  177760  166672  BIC     #177760,ANS3    ;SAVE CONDITION CODES
012114  170200                  STFPS   FPS             ;STORE FLOATING POINT STATUS
012116  022700  047504          CMP     #047504,FPS     ;CHECK FLOATING POINT STATUS
012122  001401                  BEQ     .+4             ;BRANCH IF OK
012124  104000                  HLT                     ;FPS NOT EQUAL TO 047504

012126  022767  000000  166646  CMP     #000000,ANS1    ;DID 000000 GET STORED?
```

```
012134  001401                          BEQ     .+4             ;BRANCH IF OK
012136  104002                          HLT+2                   ;ANS1 NOT EQUAL TO 000000

012140  022767  000000  166636          CMP     #000000,ANS2    ;DID 000000 GET STORED?
012146  001401                          BEQ     .+4             ;BRANCH IF OK
012150  104002                          HLT+2                   ;ANS2 NOT EQUAL TO 000000

012152  022767  000004  166626          CMP     #4,     ANS3    ;CONDITION CODES = 4?
012160  001401                          BEQ     .+4             ;BRANCH IF OK
012162  104003                          HLT+3                   ;WRONG CONDITION CODES!


                        ;*****************************************************************
                        ;TEST 111:      TEST STCFL (STORE-CONVERT FLOATING TO LONG)
                        ;       STORE   177777,177777 --> 000000,000000
                        ;       FPS = 147505,   AC = AC3,       DST = M6-R7
                        ;       FEC = 6,        FEA = FPI111
                        ;*****************************************************************

012164  104400                          SCOPE
012166  000402                          BR      TST111

012170  177777  177777  DAT111: 177777,177777

012172  170127  047500  TST111: LDFPS   #047500          ;LOAD FLOATING POINT STATUS
012200  172767  177764          LDF     DAT111, AC3     ;LOAD 177777,177777 INTO AC3
012204  175767  166572  FPI111: STCFL   AC3,    ANS1     ;STORE-CONVERT AC3 IN ANS1, ANS2
012210  013767  177776  166570          MOV     @#PS,   ANS3    ;GET CPU STATUS
012216  042767  177760  166562          BIC     #177760,ANS3    ;SAVE CONDITION CODES
012224  170200                          STFPS   FPS             ;STORE FLOATING POINT STATUS
012226  170367  166570                  STST    FEC             ;STORE EXCEPTION CODES
012232  022700  147505                  CMP     #147505,FPS     ;CHECK FLOATING POINT STATUS
012236  001401                          BEQ     .+4             ;BRANCH IF OK
012240  104000                          HLT                     ;FPS NOT EQUAL TO 147505

012242  022767  000006  166552          CMP     #6,     FEC     ;CHECK FLOATING EXCEPTION CODE
012250  001401                          BEQ     .+4             ;BRANCH IF OK
012252  104000                          HLT                     ;FEC NOT EQUAL TO 6

012254  022767  012204  166542          CMP     #FPI111, FEA    ;CHECK FLOATING EXCEPTION ADDRESS
012262  001401                          BEQ     .+4             ;BRANCH IF OK
012264  104000                          HLT                     ;FEA NOT EQUAL TO FPI111

012266  022767  000000  166506          CMP     #000000,ANS1    ;DID 000000 GET STORED?
012274  001401                          BEQ     .+4             ;BRANCH IF OK
012276  104002                          HLT+2                   ;ANS1 NOT EQUAL TO 000000

012300  022767  000000  166476          CMP     #000000,ANS2    ;DID 000000 GET STORED?
012306  001401                          BEQ     .+4             ;BRANCH IF OK
012310  104002                          HLT+2                   ;ANS2 NOT EQUAL TO 000000

012312  022767  000005  166466          CMP     #5,     ANS3    ;CONDITION CODES = 5?
012320  001401                          BEQ     .+4             ;BRANCH IF OK
012322  104003                          HLT+3                   ;WRONG CONDITION CODES!
```

```
                        ;*****************************************************************
                        ;TEST 112:      TEST STCFL (STORE-CONVERT FLOATING TO LONG)
                        ;       STORE   000200,000000 --> 000000,000000
                        ;       FPS = 047504,   AC = AC2,       DST = M6-R7
                        ;*****************************************************************

012330  104420                  SCOPE
012332  000402                  BR      TST112

012334  000200  000000  DAT112: 000200,000000

012340  170127  047500  TST112: LDFPS   #047500          ;LOAD FLOATING POINT STATUS
012344  172667  177764          LDF     DAT112, AC2     ;LOAD 000200,000000 INTO AC2
012350  175667  166432  FP1112: STCFL   AC2,    ANS1    ;STORE-CONVERT AC2 IN ANS1, ANS2
012356  013767  177776  166430          MOV     @#PS,   ANS3    ;GET CPU STATUS
012364  042767  177760  166422          BIC     #177760,ANS3    ;SAVE CONDITION CODES
012372  170200                  STFPS   FPS             ;STORE FLOATING POINT STATUS
012374  022767  047504  166404          CMP     #047504,FPS     ;CHECK FLOATING POINT STATUS
012402  001401                  BEQ     .+4             ;BRANCH IF OK
012404  104000                  HLT                     ;FPS NOT EQUAL TO 047504

012406  022767  000000  166376          CMP     #000000,ANS1    ;DID 000000 GET STORED?
012414  001401                  BEQ     .+4             ;BRANCH IF OK
012416  104002                  HLT+2                   ;ANS1 NOT EQUAL TO 000000

012420  022767  000000  166366          CMP     #000000,ANS2    ;DID 000000 GET STORED?
012426  001401                  BEQ     .+4             ;BRANCH IF OK
012430  104002                  HLT+2                   ;ANS2 NOT EQUAL TO 000000

012432  022767  000004  166356          CMP     #4,     ANS3    ;CONDITION CODES = 4?
012440  001401                  BEQ     .+4             ;BRANCH IF OK
012442  104003                  HLT+3                   ;WRONG CONDITION CODES!


                        ;*****************************************************************
                        ;TEST 113:      TEST STCFL (STORE-CONVERT FLOATING TO LONG)
                        ;       STORE   100177,177777 --> 000000,000000
                        ;       FPS = 047504,   AC = AC2,       DST = M6-R7
                        ;*****************************************************************

012434  104420                  SCOPE
012436  000402                  BR      TST113

012440  100177  177777  DAT113: 100177,177777

012444  170127  040000  TST113: LDFPS   #040000          ;LOAD FLOATING POINT STATUS
012450  172667  177764          LDF     DAT113, AC2     ;LOAD 100177,177777 INTO AC2
012454  170127  047500          LDFPS   #047500         ;LOAD FLOATING POINT STATUS
012460  175667  166316          STCFL   AC2,    ANS1    ;STORE-CONVERT AC2 IN ANS1, ANS2
012464  013767  177776  166314          MOV     @#PS,   ANS3    ;GET CPU STATUS
012472  042767  177760  166306          BIC     #177760,ANS3    ;SAVE CONDITION CODES
012500  170200                  STFPS   FPS             ;STORE FLOATING POINT STATUS
012502  022767  047504  166270          CMP     #047504,FPS     ;CHECK FLOATING POINT STATUS
012510  001401                  BEQ     .+4             ;BRANCH IF OK
012512  104003                  HLT                     ;FPS NOT EQUAL TO 047504
```

```
012512  022767  000000  166262          CMP     #000000,ANS1    ;DID 000000 GET STORED?
012520  001401                          BEQ     .+4             ;BRANCH IF OK
012522  104002                          HLT+2                   ;ANS1 NOT EQUAL TO 000000

012524  022767  000000  166252          CMP     #000000,ANS2    ;DID 000000 GET STORED?
012532  001401                          BEQ     .+4             ;BRANCH IF OK
012534  104002                          HLT+2                   ;ANS2 NOT EQUAL TO 000000

012536  022767  000004  166242          CMP     #4,     ANS3    ;CONDITION CODES = 4?
012544  001401                          BEQ     .+4             ;BRANCH IF OK
012546  104003                          HLT+3                   ;WRONG CONDITION CODES!


                        ;*****************************************************************
                        ;TEST 114:      TEST LDCLD (LOAD-CONVERT LONG TO DOUBLE)
                        ;       LOAD    000000,000000 --> 000000,000000,000000,000000
                        ;       FPS = 047704,   SRC = M6-R7,    AC = AC1
                        ;*****************************************************************

012550  104430                          SCOPE
012552  000402                          BR      TST114

012554  000000  000000          DAT114: 000000,000000

012560  170127  047700          TST114: LDFPS   #047700         ;LOAD FLOATING POINT STATUS
012564  177167  177764                  LDCLD   DAT114, AC1     ;LOAD-CONVERT 000000,000000 INTO AC1
012570  170200                          STFPS   FPS             ;STORE FLOATING POINT STATUS
012572  022700  047704                  CMP     #047704,FPS     ;CHECK FLOATING POINT STATUS
012576  001401                          BEQ     .+4             ;BRANCH IF OK
012600  104000                          HLT                     ;FPS NOT EQUAL TO 047704

012602  174167  166174                  STD     AC1,    ANS1    ;STORE AC1 IN ANS1 THRU ANS4
012606  022767  000000  166166          CMP     #000000,ANS1    ;DID 000000 GET STORED?
012614  001401                          BEQ     .+4             ;BRANCH IF OK
012616  104004                          HLT+4                   ;ANS1 NOT EQUAL TO 000000

012620  022767  000000  166156          CMP     #000000,ANS2    ;DID 000000 GET STORED?
012626  001401                          BEQ     .+4             ;BRANCH IF OK
012630  104004                          HLT+4                   ;ANS2 NOT EQUAL TO 000000

012632  022767  000000  166146          CMP     #000000,ANS3    ;DID 000000 GET STORED?
012640  001401                          BEQ     .+4             ;BRANCH IF OK
012642  104004                          HLT+4                   ;ANS3 NOT EQUAL TO 000000

012644  022767  000000  166136          CMP     #000000,ANS4    ;DID 000000 GET STORED?
012652  001401                          BEQ     .+4             ;BRANCH IF OK
012654  104004                          HLT+4                   ;ANS4 NOT EQUAL TO 000000


                        ;*****************************************************************
                        ;TEST 115:      TEST LDCLD (LOAD-CONVERT LONG TO DOUBLE)
                        ;       LOAD    125252,125252 --> 147652,125252,126000,000000
                        ;       FPS = 047710,   SRC = M6-R7,    AC = AC0
                        ;*****************************************************************

012656  104400                          SCOPE
```

```
012660  000402                          BR      TST115

012662  125252  125252          DAT115: 125252,125252

012666  170127  047700          TST115: LDFPS   #047700        ;LOAD FLOATING POINT STATUS
012672  177067  177764                  LDCLD   DAT115, AC0     ;LOAD-CONVERT 125252,125252 INTO AC0
012676  170200                          STFPS   FPS             ;STORE FLOATING POINT STATUS
012700  022700  047710                  CMP     #047710,FPS     ;CHECK FLOATING POINT STATUS
012704  001401                          BEQ     .+4             ;BRANCH IF OK
012706  104000                          HLT                     ;FPS NOT EQUAL TO 047710

012710  174067  166066                  STD     AC0,    ANS1    ;STORE AC0 IN ANS1 THRU ANS4
012714  022767  147652  166060          CMP     #147652,ANS1    ;DID 147652 GET STORED?
012722  001401                          BEQ     .+4             ;BRANCH IF OK
012724  104004                          HLT+4                   ;ANS1 NOT EQUAL TO 147652

012726  022767  125252  166050          CMP     #125252,ANS2    ;DID 125252 GET STORED?
012734  001401                          BEQ     .+4             ;BRANCH IF OK
012736  104004                          HLT+4                   ;ANS2 NOT EQUAL TO 125252

012740  022767  126000  166040          CMP     #126000,ANS3    ;DID 126000 GET STORED?
012746  001401                          BEQ     .+4             ;BRANCH IF OK
012750  104004                          HLT+4                   ;ANS3 NOT EQUAL TO 126000

012752  022767  000000  166030          CMP     #000000,ANS4    ;DID 000000 GET STORED?
012760  001401                          BEQ     .+4             ;BRANCH IF OK
012762  104004                          HLT+4                   ;ANS4 NOT EQUAL TO 000000

                        ;*****************************************************************
                        ;TEST 116:      TEST LDCLD (LOAD-CONVERT LONG TO DOUBLE.
                        ;       LOAD    052525,052525 --> 047652,125252,125000,000000
                        ;       FPS = 047700,   SRC = M6-R7,    AC = AC3
                        ;*****************************************************************

012764  104400                          SCOPE
012766  000402                          BR      TST116

012770  052525  052525          DAT116: 052525,052525

012774  170127  047700          TST116: LDFPS   #047700         ;LOAD FLOATING POINT STATUS
013000  177367  177764                  LDCLD   DAT116, AC3     ;LOAD-CONVERT 052525,052525 INTO AC3
013004  170200                          STFPS   FPS             ;STORE FLOATING POINT STATUS
013006  022700  047700                  CMP     #047700,FPS     ;CHECK FLOATING POINT STATUS
013012  001401                          BEQ     .+4             ;BRANCH IF OK
013014  104000                          HLT                     ;FPS NOT EQUAL TO 047700

013016  174367  165760                  STD     AC3,    ANS1    ;STORE AC3 IN ANS1 THRU ANS4
013022  022767  047652  165752          CMP     #047652,ANS1    ;DID 047652 GET STORED?
013030  001401                          BEQ     .+4             ;BRANCH IF OK
013032  104004                          HLT+4                   ;ANS1 NOT EQUAL TO 047652

013034  022767  125252  165742          CMP     #125252,ANS2    ;DID 125252 GET STORED?
013042  001401                          BEQ     .+4             ;BRANCH IF OK
013044  104004                          HLT+4                   ;ANS2 NOT EQUAL TO 125252
```

```
013046  022767  125000  165732          CMP     #125000,ANS3    ;DID 125000 GET STORED?
013054  001401                          BEQ     .+4             ;BRANCH IF OK
013056  104004                          HLT+4                   ;ANS3 NOT EQUAL TO 125000

013060  022767  000000  165722          CMP     #000000,ANS4    ;DID 000000 GET STORED?
013066  001401                          BEQ     .+4             ;BRANCH IF OK
013070  104004                          HLT+4                   ;ANS4 NOT EQUAL TO 000000

                        ;*****************************************************************
                        ;TEST 117:      TEST LDCLD (LOAD-CONVERT LONG TO DOUBLE)
                        ;       LOAD    000000,000001 --> 040200,000000,000000,000000
                        ;       FPS = 047700,   SRC = M6-R7,    AC = AC2
                        ;*****************************************************************

013072  104400                          SCOPE
013074  000432                          BR      TST117

013076  000000  000001          DAT117: 000000.000001

013102  170127  047700          TST117: LDFPS   #047700         ;LOAD FLOATING POINT STATUS
013106  177267  177764                  LDCLD   DAT117, AC2     ;LOAD-CONVERT 000000,000001 INTO AC2
013112  170200                          STFPS   FPS             ;STORE FLOATING POINT STATUS
013114  022700  047700                  CMP     #047700,FPS     ;CHECK FLOATING POINT STATUS
013120  001401                          BEQ     .+4             ;BRANCH IF OK
013122  104000                          HLT                     ;FPS NOT EQUAL TO 047700
013124  174267  165652                  STD     AC2,    ANS1    ;STORE AC2 IN ANS1 THRU ANS4
013130  022767  040200  165644          CMP     #040200,ANS1    ;DID 040200 GET STORED?
013136  001401                          BEQ     .+4             ;BRANCH IF OK
013140  104004                          HLT+4                   ;ANS1 NOT EQUAL TO 040200

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

013154  022767  000000  165624          CMP     #000000,ANS3    ;DID 000000 GET STORED?
013162  001401                          BEQ     .+4             ;BRANCH IF OK
013164  104004                          HLT+4                   ;ANS3 NOT EQUAL TO 000000

013166  022767  000000  165614          CMP     #000000,ANS4    ;DID 000000 GET STORED?
013174  001401                          BEQ     .+4             ;BRANCH IF OK
013176  104004                          HLT+4                   ;ANS4 NOT EQUAL TO 000000

                        ;*****************************************************************
                        ;TEST 120:      TEST LDCLD (LOAD-CONVERT LONG TO DOUBLE)
                        ;       LOAD    177777,177777 --> 140200,000000,000000,000000
                        ;       FPS = 047710,   SRC = M6-R7,    AC = AC1
                        ;*****************************************************************

013200  104400                          SCOPE
013202  000432                          BR      TST120

013204  177777  177777          DAT120: 177777,177777
```

```
013310  170127  047700          TST120: LDFPS   #047700          ;LOAD FLOATING POINT STATUS
013314  177067  177764                  LDCLD   DAT120, AC1      ;LOAD-CONVERT 177777,177777 INTO AC1
013320  170200                         STFPS   FPS              ;STORE FLOATING POINT STATUS
013322  022700  047710                 CMP     #047710,FPS      ;CHECK FLOATING POINT STATUS
013326  001401                         BEQ     .+4              ;BRANCH IF OK
013330  104000                         HLT                      ;FPS NOT EQUAL TO 047710

013332  174167  165544                 STD     AC1,    ANS1     ;STORE AC1 IN ANS1 THRU ANS4
013336  022767  140200  165536         CMP     #140200,ANS1     ;DID 140200 GET STORED?
013344  001401                         BEQ     .+4              ;BRANCH IF OK
013346  104004                         HLT+4                    ;ANS1 NOT EQUAL TO 140200

013350  022767  000000  165526         CMP     #000000,ANS2     ;DID 000000 GET STORED?
013356  001401                         BEQ     .+4              ;BRANCH IF OK
013360  104004                         HLT+4                    ;ANS2 NOT EQUAL TO 000000

013362  022767  000000  165516         CMP     #000000,ANS3     ;DID 000000 GET STORED?
013370  001401                         BEQ     .+4              ;BRANCH IF OK
013372  104004                         HLT+4                    ;ANS3 NOT EQUAL TO 000000

013374  022767  000000  165506         CMP     #000000,ANS4     ;DID 000000 GET STORED?
013402  001401                         BEQ     .+4              ;BRANCH IF OK
013404  104004                         HLT+4                    ;ANS4 NOT EQUAL TO 000000
```

```
;*****************************************************************
;TEST 121:      TEST LDCLD (LOAD-CONVERT LONG TO DOUBLE)
;       LOAD    100000.000000 --> 150000.000000.000000.000000
;       FPS = 047710,   SRC = M6-R7,    AC = AC0
;*****************************************************************
```

```
013306  104400                         SCOPE
013310  000402                         BR      TST121

013312  100000  000000          DAT121: 100000.000000

013316  170127  047700          TST121: LDFPS   #047700          ;LOAD FLOATING POINT STATUS
013322  177067  177764                  LDCLD   DAT121, AC0      ;LOAD-CONVERT 100000,000000 INTO AC0
013326  170200                         STFPS   FPS              ;STORE FLOATING POINT STATUS
013330  022700  047710                 CMP     #047710,FPS      ;CHECK FLOATING POINT STATUS
013334  001401                         BEQ     .+4              ;BRANCH IF OK
013336  104000                         HLT                      ;FPS NOT EQUAL TO 047710

013340  174067  165436                 STD     AC0,    ANS1     ;STORE AC0 IN ANS1 THRU ANS4
013344  022767  150000  165430         CMP     #150000,ANS1     ;DID 150000 GET STORED?
013352  001401                         BEQ     .+4              ;BRANCH IF OK
013354  104004                         HLT+4                    ;ANS1 NOT EQUAL TO 150000

013356  022767  000000  165420         CMP     #000000,ANS2     ;DID 000000 GET STORED?
013364  001401                         BEQ     .+4              ;BRANCH IF OK
013366  104004                         HLT+4                    ;ANS2 NOT EQUAL TO 000000

013370  022767  000000  165410         CMP     #000000,ANS3     ;DID 000000 GET STORED?
013376  001401                         BEQ     .+4              ;BRANCH IF OK
013400  104004                         HLT+4                    ;ANS3 NOT EQUAL TO 000000
```

```
013402  022767  000000  165400          CMP     #000000,ANS4    ;DID 000000 GET STORED?
013410  001401                          BEQ     .+4             ;BRANCH IF OK
013412  104004                          HLT+4                   ;ANS4 NOT EQUAL TO 000000

                                ;*****************************************************************
                                ;TEST 122:      TEST LDCLD (LOAD-CONVERT LONG TO DOUBLE)
                                ;       LOAD    077777,177777 --> 047777,177777,177000,000000
                                ;       FPS = 047700,   SRC = M6-R7,    AC = AC3
                                ;*****************************************************************

013414  104400                          SCOPE
013416  000402                          BR      TST122

013420  077777  177777          DAT122: 077777,177777

013424  170127  047700          TST122: LDFPS   #047700         ;LOAD FLOATING POINT STATUS
013430  177367  177764                  LDCLD   DAT122, AC3     ;LOAD-CONVERT 077777,177777 INTO AC3
013434  170200                          STFPS   FPS             ;STORE FLOATING POINT STATUS
013436  022700  047700                  CMP     #047700,FPS     ;CHECK FLOATING POINT STATUS
013442  001401                          BEQ     .+4             ;BRANCH IF OK
013444  104000                          HLT                     ;FPS NOT EQUAL TO 047700

013446  174367  165330                  STD     AC3,    ANS1    ;STORE AC3 IN ANS1 THRU ANS4
013452  022767  047777  165322          CMP     #047777,ANS1    ;DID 047777 GET STORED?
013460  001401                          BEQ     .+4             ;BRANCH IF OK
013462  104004                          HLT+4                   ;ANS1 NOT EQUAL TO 047777
013464  022767  177777  165312          CMP     #177777,ANS2    ;DID 177777 GET STORED?
013472  001401                          BEQ     .+4             ;BRANCH IF OK
013474  104004                          HLT+4                   ;ANS2 NOT EQUAL TO 177777

013476  022767  177000  165302          CMP     #177000,ANS3    ;DID 177000 GET STORED?
013504  001401                          BEQ     .+4             ;BRANCH IF OK
013506  104004                          HLT+4                   ;ANS3 NOT EQUAL TO 177000

013510  022767  000000  165272          CMP     #000000,ANS4    ;DID 000000 GET STORED?
013516  001401                          BEQ     .+4             ;BRANCH IF OK
013520  104004                          HLT+4                   ;ANS4 NOT EQUAL TO 000000

                                ;*****************************************************************
                                ;TEST 123:      TEST LDCLD (LOAD-CONVERT LONG TO DOUBLE)
                                ;       LOAD    000125 --> 045652,000000,000000,000000
                                ;       FPS = 047700,   SC = M2-R7,     AC = AC2
                                ;*****************************************************************

013522  104400                          SCOPE
013524  170127  047700          TST123: LDFPS   #047700         ;LOAD FLOATING POINT STATUS
013530  177227  000125                  LDCLD   #000125,AC2     ;LOAD-CONVERT 000125 INTO AC2
013534  170200                          STFPS   FPS             ;STORE FLOATING POINT STATUS
013536  022700  047700                  CMP     #047700,FPS     ;CHECK FLOATING POINT STATUS
013542  001401                          BEQ     .+4             ;BRANCH IF OK
013544  104000                          HLT                     ;FPS NOT EQUAL TO 047700

013546  174267  165230                  STD     AC2,    ANS1    ;STORE AC2 IN ANS1 THRU ANS4
```

```
013552  022767  045652  165222          CMP     #045652,ANS1    ;DID 045652 GET STORED?
013560  001401                          BEQ     .+4             ;BRANCH IF OK
013562  104004                          HLT+4                   ;ANS1 NOT EQUAL TO 045652

013564  022767  000000  165212          CMP     #000000,ANS2    ;DID 000000 GET STORED?
013572  001401                          BEQ     .+4             ;BRANCH IF OK
013574  104004                          HLT+4                   ;ANS2 NOT EQUAL TO 000000

013576  022767  000000  165202          CMP     #000000,ANS3    ;DID 000000 GET STORED?
013604  001401                          BEQ     .+4             ;BRANCH IF OK
013606  104004                          HLT+4                   ;ANS3 NOT EQUAL TO 000000

013610  022767  000000  165172          CMP     #000000,ANS4    ;DID 000000 GET STORED?
013616  001401                          BEQ     .+4             ;BRANCH IF OK
013620  104004                          HLT+4                   ;ANS4 NOT EQUAL TO 000000


                        ;*****************************************************************
                        ;TEST 124:      TEST LDCLD (LOAD-CONVERT LONG TO DOUBLE)
                        ;       LOAD    070707 --> 047743,107000,000000,000000
                        ;       FPS = 047700,   SRC = M0-R1,    AC = AC1
                        ;*****************************************************************

013622  104400                          SCOPE
013624  170127  047700          TST124: LDFPS   #047700         ;LOAD FLOATING POINT STATUS
013630  012701  070707                  MOV     #070707,R1      ;"LOAD" 070707 INTO R1
013634  177101                          LDCLD   R1,     AC1     ;LOAD-CONVERT 070707 INTO AC1
013636  170200                          STFPS   FPS             ;STORE FLOATING POINT STATUS
013640  022700  047700                  CMP     #047700,FPS     ;CHECK FLOATING POINT STATUS
013644  001401                          BEQ     .+4             ;BRANCH IF OK
013646  104000                          HLT                     ;FPS NOT EQUAL TO 047700

013650  174167  165126                  STD     AC1,    ANS1    ;STORE AC1 IN ANS1 THRU ANS4
013654  022767  047743  165120          CMP     #047743,ANS1    ;DID 047743 GET STORED?
013662  001401                          BEQ     .+4             ;BRANCH IF OK
013664  104004                          HLT+4                   ;ANS1 NOT EQUAL TO 047743

013666  022767  107000  165110          CMP     #107000,ANS2    ;DID 107000 GET STORED?
013674  001401                          BEQ     .+4             ;BRANCH IF OK
013676  104004                          HLT+4                   ;ANS2 NOT EQUAL TO 107000

013700  022767  000000  165100          CMP     #000000,ANS3    ;DID 000000 GET STORED?
013706  001401                          BEQ     .+4             ;BRANCH IF OK
013710  104004                          HLT+4                   ;ANS3 NOT EQUAL TO 000000

013712  022767  000000  165070          CMP     #000000,ANS4    ;DID 000000 GET STORED?
013720  001401                          BEQ     .+4             ;BRANCH IF OK
013722  104004                          HLT+4                   ;ANS4 NOT EQUAL TO 000000


                        ;*****************************************************************
                        ;TEST 125:      TEST LDCLD (LOAD-CONVERT LONG TO DOUBLE)
                        ;       LOAD    107070,107070 --> 147743,107343,110000,000000
                        ;       FPS = 047710,   SRC = M6-R7,    AC = AC3
                        ;*****************************************************************
```

```
013724  104400                          SCOPE
013726  000402                          BR      TST125

013730  107070  107070          DAT125: 107070,107070

013734  170127  047700          TST125: LDFPS   #047700         ;LOAD FLOATING POINT STATUS
013740  177367  177764                  LDCLD   DAT125, AC3     ;LOAD-CONVERT 107070,107070 INTO AC3
013744  170200                          STFPS   FPS             ;STORE FLOATING POINT STATUS
013746  022700  047710                  CMP     #047710,FPS     ;CHECK FLOATING POINT STATUS
013752  001401                          BEQ     .+4             ;BRANCH IF OK
013754  104000                          HLT                     ;FPS NOT EQUAL TO 047710

013756  174367  165020                  STD     AC3,    ANS1    ;STORE AC3 IN ANS1 THRU ANS4
013762  022767  147743  165012          CMP     #147743,ANS1    ;DID 147743 GET STORED?
013770  001401                          BEQ     .+4             ;BRANCH IF OK
013772  104004                          HLT+4                   ;ANS1 NOT EQUAL TO 147743

013774  022767  107343  165002          CMP     #107343,ANS2    ;DID 107343 GET STORED?
014002  001401                          BEQ     .+4             ;BRANCH IF OK
014004  104004                          HLT+4                   ;ANS2 NOT EQUAL TO 107343

014006  022767  110000  164772          CMP     #110000,ANS3    ;DID 110000 GET STORED?
014014  001401                          BEQ     .+4             ;BRANCH IF OK
014016  104004                          HLT+4                   ;ANS3 NOT EQUAL TO 110000

014020  022767  000000  164762          CMP     #000000,ANS4    ;DID 000000 GET STORED?
014026  001401                          BEQ     .+4             ;BRANCH IF OK
014030  104004                          HLT+4                   ;ANS4 NOT EQUAL TO 000000
```

```
                        ;*****************************************************************
                        ;TEST 126:      TEST STCDL (STORE-CONVERT DOUBLE TO LONG)
                        ;       STORE   000000,000000,000000,000000 --> 000000,000000
                        ;       FPS = 047704,   AC = AC2,       DST = M6-R7
                        ;*****************************************************************

014032  104400                          SCOPE
014034  000404                          BR      TST126

014036  000000  000000  000000  DAT126: 000000,000000,000000,000000
014044  000000

014046  170127  047700          TST126: LDFPS   #047700         ;LOAD FLOATING POINT STATUS
014052  172667  177760                  LDD     DAT126, AC2     ;LOAD 000000,000000,000000,000000 INTO AC2
014056  175667  164720                  STCDL   AC2,    ANS1    ;STORE-CONVERT AC2 IN ANS1, ANS2
014062  013767  177776  164716          MOV     @#PS,   ANS3    ;GET CPU STATUS
014070  042767  177760  164710          BIC     #177760,ANS3    ;SAVE CONDITION CODES
014076  170200                          STFPS   FPS             ;STORE FLOATING POINT STATUS
014100  022700  047704                  CMP     #047704,FPS     ;CHECK FLOATING POINT STATUS
014104  001401                          BEQ     .+4             ;BRANCH IF OK
014106  104000                          HLT                     ;FPS NOT EQUAL TO 047704

014110  022767  000000  164664          CMP     #000000,ANS1    ;DID 000000 GET STORED?
014116  001401                          BEQ     .+4             ;BRANCH IF OK
014120  104002                          HLT+2                   ;ANS1 NOT EQUAL TO 000000
```

```
014122  022767  000000  164654          CMP     #000000,ANS2    ;DID 000000 GET STORED?
014130  001401                          BEQ     .+4             ;BRANCH IF OK
014132  104002                          HLT+2                   ;ANS2 NOT EQUAL TO 000000

014134  022767  000004  164644          CMP     #4,     ANS3    ;CONDITION CODES = 4?
014142  001401                          BEQ     .+4             ;BRANCH IF OK
014144  104003                          HLT+3                   ;WRONG CONDITION CODES!


                        ;*****************************************************************
                        ;TEST 127:      TEST STCDL (STORE-CONVERT DOUBLE TO LONG)
                        ;       STORE   041252,125252,125252,125252 --> 000000
                        ;       FPS = 047700,   AC = AC1,       DST = M2-R7
                        ;*****************************************************************

014146  104400                          SCOPE
014150  000404                          BR      TST127

014152  041252  125252  125252  DAT127: 041252,125252,125252,125252
014160  125252

014162  170127  047700          TST127: LDFPS   #047700         ;LOAD FLOATING POINT STATUS
014166  172567  177760                  LDD     DAT127, AC1     ;LOAD 041252,125252,125252,125252 INTO AC1
014172  175527                          STCDL   AC1,    (PC)+   ;STORE-CONVERT AC1 IN .+2
014174  000000                  ANR127: 0
014176  000402                          BR      .+6
014200  000000                          HALT                    ;PC FAILURE
014202  000000                          HALT
014204  013767  177776  164572          MOV     @#PS,   ANS2    ;GET CPU STATUS
014212  042767  177760  164564          BIC     #177760,ANS2    ;SAVE CONDITION CODES
014220  016767  177750  164554          MOV     ANR127, ANS1    ;SAVE THE ANSWER
014226  170200                          STFPS   FPS             ;STORE FLOATING POINT STATUS
014230  022700  047700                  CMP     #047700,FPS     ;CHECK FLOATING POINT STATUS
014234  001401                          BEQ     .+4             ;BRANCH IF OK
014236  104000                          HLT                     ;FPS NOT EQUAL TO 047700

014240  022767  000000  164534          CMP     #000000,ANS1    ;DID 000000 GET STORED?
014246  001401                          BEQ     .+4             ;BRANCH IF OK
014250  104001                          HLT+1                   ;ANS1 NOT EQUAL TO 000000

014252  022767  000000  164524          CMP     #0,     ANS2    ;CONDITION CODES = 0?
014260  001401                          BEQ     .+4             ;BRANCH IF OK
014262  104002                          HLT+2                   ;WRONG CONDITION CODES!


                        ;*****************************************************************
                        ;TEST 130:      TEST STCDL (STORE-CONVERT DOUBLE TO LONG)
                        ;       STORE   141252,125252,125252,125252 --> 177777
                        ;       FPS = 047710,   AC = AC3,       DST = M0-R0
                        ;*****************************************************************

014264  104400                          SCOPE
014266  000404                          BR      TST130

014270  141252  125252  125252  DAT130: 141252,125252,125252,125252
014276  125252
```

```
014300  170127  047700          TST130: LDFPS   #047700          ;LOAD FLOATING POINT STATUS
014304  172767  177760                  LDD     DAT130, AC3      ;LOAD 141252,125252,125252,125252 INTO AC3
014310  175700                          STCDL   AC3,    R0       ;STORE-CONVERT AC3 IN R0
014312  013767  177776  164464          MOV     @#PS,   ANS2     ;GET CPU STATUS
014320  042767  177760  164456          BIC     #177760,ANS2     ;SAVE CONDITION CODES
014326  010067  164450                  MOV     R0,     ANS1     ;SAVE R0
014332  170200                          STFPS   FPS              ;STORE FLOATING POINT STATUS
014334  022700  047710                  CMP     #047710,FPS      ;CHECK FLOATING POINT STATUS
014340  001401                          BEQ     .+4              ;BRANCH IF OK
014342  104000                          HLT                      ;FPS NOT EQUAL TO 047710

014344  022767  177777  164430          CMP     #177777,ANS1     ;DID 177777 GET STORED?
014352  001401                          BEQ     .+4              ;BRANCH IF OK
014354  104001                          HLT+1                    ;ANS1 NOT EQUAL TO 177777

014356  022767  000010  164420          CMP     #10.    ANS2     ;CONDITION CODES = 10?
014364  001401                          BEQ     .+4              ;BRANCH IF OK
014366  104002                          HLT+2                    ;WRONG CONDITION CODES!


                                ;*****************************************************************
                                ;TEST 131:      TEST STCDL (STORE-CONVERT DOUBLE TO LONG)
                                ;       STORE   040177,177777,177777,177777 --> 000000,000000
                                ;       FPS = 047704,   AC = AC2,       DST = M6-R7
                                ;*****************************************************************

014370  104400                          SCOPE
014372  000404                          BR      TST131

014374  040177  177777  177777  DAT131: 040177,177777,177777,177777
014402  177777

014404  170127  047700          TST131: LDFPS   #047700          ;LOAD FLOATING POINT STATUS
014410  172667  177760                  LDD     DAT131, AC2      ;LOAD 040177,177777,177777,177777 INTO AC2
014414  175667  164362                  STCDL   AC2,    ANS1     ;STORE-CONVERT AC2 IN ANS1, ANS2
014420  013767  177776  164360          MOV     @#PS,   ANS3     ;GET CPU STATUS
014426  042767  177760  164352          BIC     #177760,ANS3     ;SAVE CONDITION CODES
014434  170200                          STFPS   FPS              ;STORE FLOATING POINT STATUS
014436  022700  047704                  CMP     #047704,FPS      ;CHECK FLOATING POINT STATUS
014442  001401                          BEQ     .+4              ;BRANCH IF OK
014444  104000                          HLT                      ;FPS NOT EQUAL TO 047704

014446  022767  000000  164326          CMP     #000000,ANS1     ;DID 000000 GET STORED?
014454  001401                          BEQ     .+4              ;BRANCH IF OK
014456  104002                          HLT+2                    ;ANS1 NOT EQUAL TO 000000

014460  022767  000000  164316          CMP     #000000,ANS2     ;DID 000000 GET STORED?
014466  001401                          BEQ     .+4              ;BRANCH IF OK
014470  104002                          HLT+2                    ;ANS2 NOT EQUAL TO 000000

014472  022767  000004  164306          CMP     #4,     ANS3     ;CONDITION CODES = 4?
014500  001401                          BEQ     .+4              ;BRANCH IF OK
014502  104003                          HLT+3                    ;WRONG CONDITION CODES!
```

```
                        ;*****************************************************************
                        ;TEST 132:      TEST STCDL (STORE-CONVERT DOUBLE TO LONG)
                        ;       STORE   140177,177777,177777,177777 --> 000000,000000
                        ;       FPS = 047704,   AC = AC0,       DST = M6-R7
                        ;*****************************************************************

014504  104400                  SCOPE
014506  000404                  BR      TST132

014510  140177  177777  177777  DAT132: 140177,177777,177777,177777
014516  177777

014520  170127  047700          TST132: LDFPS   #047700         ;LOAD FLOATING POINT STATUS
014524  172467  177760                  LDD     DAT132, AC0     ;LOAD 140177,177777,177777,177777 INTO AC0
014530  175467  164246                  STCDL   AC0,    ANS1    ;STORE-CONVERT AC0 IN ANS1, ANS2
014534  013767  177776  164244          MOV     @#PS,   ANS3    ;GET CPU STATUS
014542  042767  177760  164236          BIC     #177760,ANS3    ;SAVE CONDITION CODES
014550  170200                  STFPS   FPS             ;STORE FLOATING POINT STATUS
014552  022700  047704                  CMP     #047704,FPS     ;CHECK FLOATING POINT STATUS
014556  001401                  BEQ     .+4             ;BRANCH IF OK
014560  104000                  HLT                     ;FPS NOT EQUAL TO 047704

014562  022767  000000  164212          CMP     #000000,ANS1    ;DID 000000 GET STORED?
014570  001401                  BEQ     .+4             ;BRANCH IF OK
014572  104002                  HLT+2                   ;ANS1 NOT EQUAL TO 000000

014574  022767  000000  164202          CMP     #000000,ANS2    ;DID 000000 GET STORED?
014602  001401                  BEQ     .+4             ;BRANCH IF OK
014604  104002                  HLT+2                   ;ANS2 NOT EQUAL TO 000000

014606  022767  000004  164172          CMP     #4,     ANS3    ;CONDITION CODES = 4?
014614  001401                  BEQ     .+4             ;BRANCH IF OK
014616  104003                  HLT+3                   ;WRONG CONDITION CODES!


                        ;*****************************************************************
                        ;TEST 133:      TEST STCDL (STORE-CONVERT DOUBLE TO LONG)
                        ;       STORE   040200,000001,000001,000001 --> 000000,000001
                        ;       FPS = 047700,   AC = AC3,       DST = M6-R7
                        ;*****************************************************************

014620  104400                  SCOPE
014622  000404                  BR      TST133

014624  040200  000001  000001  DAT133: 040200,000001,000001,000001
014632  000001

014634  170127  047700          TST133: LDFPS   #047700         ;LOAD FLOATING POINT STATUS
014640  172767  177760                  LDD     DAT133, AC3     ;LOAD 040200,000001,000001,000001 INTO AC3
014644  175767  164132                  STCDL   AC3,    ANS1    ;STORE-CONVERT AC3 IN ANS1, ANS2
014650  013767  177776  164130          MOV     @#PS,   ANS3    ;GET CPU STATUS
014656  042767  177760  164122          BIC     #177760,ANS3    ;SAVE CONDITION CODES
014664  170200                  STFPS   FPS             ;STORE FLOATING POINT STATUS
014666  022700  047700                  CMP     #047700,FPS     ;CHECK FLOATING POINT STATUS
014672  001401                  BEQ     .+4             ;BRANCH IF OK
014674  104000                  HLT                     ;FPS NOT EQUAL TO 047700
```

```
014676  022767  000000  164076          CMP     #000000,ANS1    ;DID 000000 GET STORED?
014704  001401                          BEQ     .+4             ;BRANCH IF OK
014706  104002                          HLT+2                   ;ANS1 NOT EQUAL TO 000000

014710  022767  000001  164066          CMP     #000001,ANS2    ;DID 000001 GET STORED?
014716  001401                          BEQ     .+4             ;BRANCH IF OK
014720  104002                          HLT+2                   ;ANS2 NOT EQUAL TO 000001

014722  022767  000000  164056          CMP     #0,     ANS3    ;CONDITION CODES = 0?
014730  001401                          BEQ     .+4             ;BRANCH IF OK
014732  104003                          HLT+3                   ;WRONG CONDITION CODES!


                                ;*****************************************************************
                                ;TEST 134:      TEST STCDL (STORE-CONVERT DOUBLE TO LONG)
                                ;       STORE   140200,000001,000001,000001 --> 177777,177777
                                ;       FPS = 04771C,   AC = AC1,       DST = M6-R7
                                ;*****************************************************************

014734  104400                          SCOPE
014736  000404                          BR      TST134

014740  140200  000001  000001  DAT134: 140200,000001,000001,000001
014746  000001

014750  170127  047700          TST134: LDFPS   #047700         ;LOAD FLOATING POINT STATUS
014754  172567  177760                  LDD     DAT134, AC1     ;LOAD 140200,000001,000001,000001 INTO AC1
014760  175567  164016                  STCDL   AC1,    ANS1    ;STORE-CONVERT AC1 IN ANS1, ANS2
014764  013767  177776  164014          MOV     @#PS,   ANS3    ;GET CPU STATUS
014772  042767  177760  164006          BIC     #177760,ANS3    ;SAVE CONDITION CODES
015000  170200                          STFPS   FPS             ;STORE FLOATING POINT STATUS
015002  022700  047710                  CMP     #047710,FPS     ;CHECK FLOATING POINT STATUS
015006  001401                          BEQ     .+4             ;BRANCH IF OK
015010  104000                          HLT                     ;FPS NOT EQUAL TO 047710

015012  022767  177777  163762          CMP     #177777,ANS1    ;DID 177777 GET STORED?
015020  001401                          BEQ     .+4             ;BRANCH IF OK
015022  104002                          HLT+2                   ;ANS1 NOT EQUAL TO 177777

015024  022767  177777  163752          CMP     #177777,ANS2    ;DID 177777 GET STORED?
015032  001401                          BEQ     .+4             ;BRANCH IF OK
015034  104002                          HLT+2                   ;ANS2 NOT EQUAL TO 177777

015036  022767  000010  163742          CMP     #10,    ANS3    ;CONDITION CODES = 10?
015044  001401                          BEQ     .+4             ;BRANCH IF OK
015046  104003                          HLT+3                   ;WRONG CONDITION CODES!


                                ;*****************************************************************
                                ;TEST 135:      TEST STCDL (STORE-CONVERT DOUBLE TO LONG)
                                ;       STORE   047777,177777,177777,177777 --> 077777,177777
                                ;       FPS = 047700,   AC = ACC,       DST = M6-R7
                                ;*****************************************************************

015050  104400                          SCOPE
```

```
015052  000404                          BR      TST135

015054  047777  177777  177777  DAT135: 047777,177777,177777,177777
015062  177777

015064  170127  047700          TST135: LDFPS   #047700         ;LOAD FLOATING POINT STATUS
015070  172467  177760                  LDD     DAT135, AC0     ;LOAD 047777,177777,177777,177777 INTO AC0
015074  175467  163702                  STCDL   AC0,    ANS1    ;STORE-CONVERT AC0 IN ANS1, ANS2
015100  013767  177776  163700          MOV     @#PS,   ANS3    ;GET CPU STATUS
015106  042767  177760  163672          BIC     #177760,ANS3    ;SAVE CONDITION CODES
015114  170200                          STFPS   FPS             ;STORE FLOATING POINT STATUS
015116  022700  047700                  CMP     #047700,FPS     ;CHECK FLOATING POINT STATUS
015122  001401                          BEQ     .+4             ;BRANCH IF OK
015124  104000                          HLT                     ;FPS NOT EQUAL TO 047700

015126  022767  077777  163646          CMP     #077777,ANS1    ;DID 077777 GET STORED?
015134  001401                          BEQ     .+4             ;BRANCH IF OK
015136  104002                          HLT+2                   ;ANS1 NOT EQUAL TO 077777

015140  022767  177777  163636          CMP     #177777,ANS2    ;DID 177777 GET STORED?
015146  001401                          BEQ     .+4             ;BRANCH IF OK
015150  104002                          HLT+2                   ;ANS2 NOT EQUAL TO 177777

015152  022767  000000  163626          CMP     #0,     ANS3    ;CONDITION CODES = 0?
015160  001401                          BEQ     .+4             ;BRANCH IF OK
015162  104003                          HLT+3                   ;WRONG CONDITION CODES!
```

```
;*****************************************************************
;TEST 136:      TEST STCDL (STORE-CONVERT DOUBLE TO LONG)
;       STORE   050000,000000,000000,000000 --> 000000,000000
;       FPS = 147705,   AC = AC2,       DST = M6-R7
;*****************************************************************

015164  104400                          SCOPE
015166  000404                          BR      TST136

015170  050000  000000  000000  DAT136: 050000,000000,000000,000000
015176  000000

015200  170127  047700          TST136: LDFPS   #047700         ;LOAD FLOATING POINT STATUS
015204  172667  177760                  LDD     DAT136, AC2     ;LOAD 050000,000000,000000,000000 INTO AC2
015210  175667  163566                  STCDL   AC2,    ANS1    ;STORE-CONVERT AC2 IN ANS1, ANS2
015214  013767  177776  16[illegible]   MOV     @#PS,   ANS3    ;GET CPU STATUS
015222  042767  177760  16[illegible]   BIC     #177760,ANS3    ;SAVE CONDITION CODES
015230  170200                          STFPS   FPS             ;STORE FLOATING POINT STATUS
015232  170367  163564                  STST    FEC             ;STORE EXCEPTION CODES
015236  022700  147705                  CMP     #147705,FPS     ;CHECK FLOATING POINT STATUS
015242  001401                          BEQ     .+4             ;BRANCH IF OK
015244  104000                          HLT                     ;FPS NOT EQUAL TO 147705

015246  022767  000006  163546          CMP     #6,     FEC     ;CHECK FLOATING EXCEPTION CODE
015254  001401                          BEQ     .+4             ;BRANCH IF OK
015256  104000                          HLT                     ;FEC NOT EQUAL TO 6

015260  022767  000000  163514          CMP     #000000,ANS1    ;DID 000000 GET STORED?
```

```
                        ;TEST OF .DCVL, STCDL
                        ;ES* SECTION

015266  001401                  BEQ     .+4             ;BRANCH IF OK
015270  104002                  HLT+2                   ;ANS1 NOT EQUAL TO 000000

015272  022767  000000  163504  CMP     #000000,ANS2    ;DID 000000 GET STORED*
015300  001401                  BEQ     .+4             ;BRANCH IF OK
015302  104002                  HLT+2                   ;ANS2 NOT EQUAL TO 000000

015304  022767  000005  163474  CMP     #5,     ANS3    ;CONDITION CODES = 5*
015312  001401                  BEQ     .+4             ;BRANCH IF OK
015314  104003                  HLT+3                   ;WRONG CONDITION CODES!


                        ;*****************************************************************
                        ;TEST 137:      TEST STCDL (STORE-CONVERT DOUBLE TO LONG.
                        ;       STORE   150000.000000,000377.177777 --> 150000.000000
                        ;       FPS = 047710,   AC = AC0,       DST = M6-R7
                        ;*****************************************************************

015316  104400                  SCOPE
015320  000434                  BR      *ST137

015322  150000  000000  000377  DAT137: 150000.000000,000377.177777
015330  177777

015332  170127  047700          ST137:  LDFPS   #047700         ;LOAD FLOATING POINT STATUS
015336  172467  177760                  LDD     DAT137, AC0     ;LOAD 150000.000000,000377.177777 INTO AC0
015342  175467  163434                  STCDL   AC0,    ANS1    ;STORE-CONVERT AC0 IN ANS1, ANS2
015346  013767  177776  163432          MOV     @#PS,   ANS3    ;GET CPU STATUS
015354  042767  177760  163424          BIC     #177760,ANS3    ;SAVE CONDITION CODES
015362  170200                  STFPS   FPS             ;STORE FLOATING POINT STATUS
015364  022700  047710                  CMP     #047710,FPS     ;CHECK FLOATING POINT STATUS
015370  001401                  BEQ     .+4             ;BRANCH IF OK
015372  104000                  HLT                     ;FPS NOT EQUAL TO 047710

015374  022767  100000  163400          CMP     #100000,ANS1    ;DID 100000 GET STORED*
015402  001401                  BEQ     .+4             ;BRANCH IF OK
015404  104002                  HLT+2                   ;ANS1 NOT EQUAL TO 100000

015406  022767  000000  163370          CMP     #000000,ANS2    ;DID 000000 GET STORED*
015414  001401                  BEQ     .+4             ;BRANCH IF OK
015416  104002                  HLT+2                   ;ANS2 NOT EQUAL TO 000000

015420  022767  000010  163360          CMP     #10.,   ANS3    ;CONDITION CODES = 10*
015426  001401                  BEQ     .+4             ;BRANCH IF OK
015430  104003                  HLT+3                   ;WRONG CONDITION CODES!


                        ;*****************************************************************
                        ;TEST 140:      TEST STCDL (STORE-CONVERT DOUBLE TO LONG.
                        ;       STORE   150000.000000,000400.000000 --> 000000.000000
                        ;       FPS = 147705,   AC = AC3,       DST = M6-R*
                        ;*****************************************************************

015432  104400                  SCOPE
015434  000434                  BR      *ST140
```

```
                        ;TEST SECTION

015436  150000  000000  000400  DAT140: 150000.000000,000400,000000
015444  000000

015446  170127  047700          ;ST140: LDFPS   #047700         ;LOAD FLOATING POINT STATUS
015452  172567  177766                  LDD     DAT140, AC3     ;LOAD 150000,000000,000400,000000 INTO AC3
015456  175767  163320                  STCDL   AC3,    ANS1    ;STORE-CONVERT AC3 IN ANS1, ANS2
015462  013767  177776  163316          MOV     #PS,    ANS3    ;GET CPU STATUS
015470  042767  177760  163312          BIC     #177760,ANS3    ;SAVE CONDITION CODES
015476  170200                  STFPS   FPS             ;STORE FLOATING POINT STATUS
015500  175367  163316                  STST    FEC             ;STORE EXCEPTION CODES
015506  022700  147705                  CMP     #147705,FPS     ;CHECK FLOATING POINT STATUS
015512  001401                  BEQ     .+4             ;BRANCH IF OK
015514  104000                  HLT                     ;FPS NOT EQUAL TO 147705

015516  022767  000006  163300          CMP     #6,     FEC     ;CHECK FLOATING EXCEPTION CODE
015524  001401                  BEQ     .+4             ;BRANCH IF OK
015526  104000                  HLT                     ;FEC NOT EQUAL TO 6

015530  022767  000000  163246          CMP     #000000,ANS1    ;DID 000000 GET STORED?
015536  001401                  BEQ     .+4             ;BRANCH IF OK
015540  104002                  HLT+2                   ;ANS1 NOT EQUAL TO 000000

015542  022767  000000  163236          CMP     #000000,ANS2    ;DID 000000 GET STORED?
015550  001401                  BEQ     .+4             ;BRANCH IF OK
015552  104002                  HLT+2                   ;ANS2 NOT EQUAL TO 000000

015554  022767  000005  163226          CMP     #5,     ANS3    ;CONDITION CODES = 5?
015562  001401                  BEQ     .+4             ;BRANCH IF OK
015564  104003                  HLT+3                   ;WRONG CONDITION CODES!


                        ;*****************************************************************
                        ;TEST 141:      TEST STCDL (STORE-CONVERT DOUBLE TO LONG
                        ;       STORE   100000.000000,000000.000000 --> 000000.000000
                        ;       FPS = 047704,   AC = AC1,       DST = @R5-R7
                        ;*****************************************************************
015566  104400                  SCOPE
015570  000404                  BR      TST141

015572  100000  000000  000000  DAT141: 100000.000000,000000.000000
015600  000000

015602  170127  040200          ;TST141: LDFPS   #040200         ;LOAD FLOATING POINT STATUS
015606  172567  177760                  LDD     DAT141, AC1     ;LOAD 100000,000000,000000,000000 INTO AC1
015612  170127  047700                  LDFPS   #047700         ;LOAD FLOATING POINT STATUS
015616  175567  163162                  STCDL   AC1,    ANS1    ;STORE-CONVERT AC1 IN ANS1, ANS2
015622  013767  177776  163160          MOV     #PS,    ANS3    ;GET CPU STATUS
015630  042767  177760  163152          BIC     #177760,ANS3    ;SAVE CONDITION CODES
015636  170200                  STFPS   FPS             ;STORE FLOATING POINT STATUS
015640  022700  047704                  CMP     #047704,FPS     ;CHECK FLOATING POINT STATUS
015644  001401                  BEQ     .+4             ;BRANCH IF OK
015646  104000                  HLT                     ;FPS NOT EQUAL TO 047704

015650  022767  000000  163126          CMP     #000000,ANS1    ;DID 000000 GET STORED?
015656  001401                  BEQ     .+4             ;BRANCH IF OK
015660  104002                  HLT+2                   ;ANS1 NOT EQUAL TO 000000
```

```
015662  022767  000000  163116          CMP     #000000,ANS2    ;DID 000000 GET STORED?
015666  001401                          BEQ     .+4             ;BRANCH IF OK
015670  104002                          HLT+2                   ;ANS2 NOT EQUAL TO 000000

015672  022767  000004  163106          CMP     #4,     ANS3    ;CONDITION CODES = 4?
015676  001401                          BEQ     .+4             ;BRANCH IF OK
015700  104003                          HLT+3                   ;WRONG CONDITION CODES!

                        ;*****************************************************************
                        ;TEST 142:      TEST STCDL (STORE-CONVERT DOUBLE TO LONG.
                        ;       STORE   177777.177777, 177777.177777 --> 000000.000000
                        ;       FPS = 147705,   AC = AC3,       DST = M6-R7
                        ;*****************************************************************

015704  104420                          SCOPE
015706  000454                          BR      TST142

015710  177777  177777  177777  DAT142: 177777.177777.177777.177777
015716  177777

015720  170127  047700          TST142: LDFPS   #047700         ;LOAD FLOATING POINT STATUS
015724  172767  177760                  LDD     DAT142, AC3     ;LOAD 177777,177777,177777,177777 INTO AC3
015730  175767  163046                  STCDL   AC3,    ANS1    ;STORE-CONVERT AC3 IN ANS1, ANS2
015734  013767  177776  163044          MOV     @#PS,   ANS3    ;GET CPU STATUS
015742  042767  177760  163036          BIC     #177760,ANS3    ;SAVE CONDITION CODES
015750  170200                          STFPS   FPS             ;STORE FLOATING POINT STATUS
015752  170367  163044                  STST    FEC             ;STORE EXCEPTION CODES
015756  022700  147705                  CMP     #147705,FPS     ;CHECK FLOATING POINT STATUS
015762  001401                          BEQ     .+4             ;BRANCH IF OK
015764  104000                          HLT                     ;FPS NOT EQUAL TO 147705

015766  022767  000006  163026          CMP     #6,     FEC     ;CHECK FLOATING EXCEPTION CODE
015774  001401                          BEQ     .+4             ;BRANCH IF OK
015776  104000                          HLT                     ;FEC NOT EQUAL TO 6

016000  022767  000000  162774          CMP     #000000,ANS1    ;DID 000000 GET STORED?
016006  001401                          BEQ     .+4             ;BRANCH IF OK
016010  104002                          HLT+2                   ;ANS1 NOT EQUAL TO 000000

016012  022767  000000  162764          CMP     #000000,ANS2    ;DID 000000 GET STORED?
016020  001401                          BEQ     .+4             ;BRANCH IF OK
016022  104002                          HLT+2                   ;ANS2 NOT EQUAL TO 000000

016024  022767  000005  162754          CMP     #5,     ANS3    ;CONDITION CODES = 5?
016032  001401                          BEQ     .+4             ;BRANCH IF OK
016034  104003                          HLT+3                   ;WRONG CONDITION CODES!

                        ;*****************************************************************
                        ;TEST 143:      TEST STCDL (STORE-CONVERT DOUBLE TO LONG.
                        ;       STORE   100177.177777,177777.177777 --> 000000.000000
                        ;       FPS = 047704,   AC = AC3,       DST = M6-R7
                        ;*****************************************************************

016036  104420                          SCOPE
```

```
016040  000404                          BR      TST143

016042  100177  177777  177777  DAT143: 100177,177777,177777,177777
016050  177777

016052  170127  040200          TST143: LDFPS   #040200        ;LOAD FLOATING POINT STATUS
016056  172767  177760                  LDD     DAT143, AC3     ;LOAD 100177,177777,177777,177777 INTO AC3
016062  170127  047700                  LDFPS   #047700         ;LOAD FLOATING POINT STATUS
016066  175767  162710                  STCDL   AC3,    ANS1    ;STORE-CONVERT AC3 IN ANS1, ANS2
016072  013767  177776  162706        MOV     @#PS,   ANS3    ;GET CPU STATUS
016100  042767  177760  162700        BIC     #177760,ANS3    ;SAVE CONDITION CODES
016106  170200                        STFPS   FPS             ;STORE FLOATING POINT STATUS
016110  022700  047704                CMP     #047704,FPS     ;CHECK FLOATING POINT STATUS
016114  001401                        BEQ     .+4             ;BRANCH IF OK
016116  104000                        HLT                     ;FPS NOT EQUAL TO 047704

016120  022767  000000  162654        CMP     #000000,ANS1    ;DID 000000 GET STORED?
016126  001401                        BEQ     .+4             ;BRANCH IF OK
016130  104002                        HLT+2                   ;ANS1 NOT EQUAL TO 000000

016132  022767  000000  162644        CMP     #000000,ANS2    ;DID 000000 GET STORED?
016140  001401                        BEQ     .+4             ;BRANCH IF OK
016142  104002                        HLT+2                   ;ANS2 NOT EQUAL TO 000000

016144  022767  000004  162634        CMP     #4,     ANS3    ;CONDITION CODES = 4?
016152  001401                        BEQ     .+4             ;BRANCH IF OK
016154  104003                        HLT+3                   ;WRONG CONDITION CODES!
```

```
016156  104400                  DONE:   SCOPE
016160  032737  002000  177570          BIT     #SW10,@#SWR     ;PING THE BELL?
016166  001005                          BNE     1$              ;NO!
016170  012767  002207  001242          MOV     #BELL,.TYPE     ;TYPE A BELL
016176  000004  017440                  TYPE    ,.TYPE
016202  005046                  1$:     CLR     -(6)            ;CLEAR TRACE TRAP
016204  032737  010000  177570          BIT     #SW12,@#SWR     ;RUN WITH TRT?
016212  001010                          BNE     2$
016214  005167  001222                  COM     TRPB
016220  100005                          BPL     2$
016222  052716  000020                  BIS     #20,(6)         ;SET TRACE TRAP
016226  012746  001062                  MOV     #BEGIN,-(6)     ;JUMP TO START OF TEST
016232  000412                          BR      YESRT
016234  012746  001062          3$:     MOV     #BEGIN,-(6)     ;JUMP TO START OF TEST
016240  013700  000042                  MOV     @#42,R0         ;GET MONITOR ADDRESS
016244  001404                          BEQ     3$              ;IF NONE
016246  004710                          JSR     7,(0)           ;GO TO MONITOR
016250  000240                          NOP
016252  000240                          NOP
016254  000240                          NOP
016256  000002                  3$:     RTI
016260  000002                  YESRT:  RTI                     ;RETURN TO PROGRAM FROM TRAP

016262  032737  000400  177570  .EMT:   BIT     #SW08,@#SWR     ;KILL LOOP OR LOOP ON SPEC. TEST
016270  001404                          BEQ     1$
016272  123767  177570  162500          CMPB    @#SWR,ICNT      ;ON RIGHT TEST?  #SW7-0#
016300  001437                          BEQ     OVER
016302  113703  177570          1$:     MOVB    @#SWR,R3        ;GET UB BITS
016306  170003                          LDUB
016310  032737  040000  177570          BIT     #SW14,@#SWR     ;LOOP ON TEST
016316  001026                          BNE     KIT
016320  032737  004000  177570          BIT     #SW11,@#SWR     ;KILL ITERATIONS
016326  001012                          BNE     SAVLAD
016330  105767  162445                  TSTB    ICNT+1
016334  001404                          BEQ     2$              ;BRANCH IF FIRST
016336  126767  001106  162435          CMPB    TIMES,ICNT+1    ;DONE?
016344  001013                          BNE     KIT             ;BRANCH IF NOT
016346  112767  000001  162425  2$:     MOVB    #1,ICNT+1       ;FIRST ITERATION
016354  105267  162420          SAVLAD: INCB    ICNT            ;COUNT TEST NUMBERS
016360  011667  001060                  MOV     (6),LAD         ;SAVE LOOP ADDRESS
016364  016737  162410  177570          MOV     ICNT,@#DISPLAY  ;DISPLAY TEST NO. AND ITERATION COUNT
016372  000002                          RTI                     ;RETURN

016374  105267  162401          KIT:    INCB    ICNT+1
016400  016737  162374  177570  OVER:   MOV     ICNT,@#DISPLAY  ;SET UP DISPLAY
016406  005767  001032                  TST     LAD             ;FIRST ONE?
016412  001760                          BEQ     SAVLAD
016414  016716  001024                  MOV     LAD,(6)         ;BUDGE RETURN ADDRESS
016420  000002                          RTI                     ;TAKES PS
```

```
016422  032737  002000  177570  .TRP:   BIT     #SW10,@#SWR     ;BELL ON ERROR?
016430  001405                          BEQ     1$              ;NO - SKIP
016432  012767  000207  001000          MOV     #BELL,.TYPE     ;TYPE A BELL
016440  000004  017440                  TYPE    .TYPE
016444  004767  000406          1$:     JSR     PC,ERROR        ;COUNT THE NUMBER OF ERRORS
016450  010446                          MOV     R4,-(6)
016452  032737  020000  177570          BIT     #SW13,@#SWR     ;SKIP TYPEOUT IF SET
016460  001072                          BNE     4$
016462  000004  017406                  TYPE    ,RETURN
016466  016646  000002                  MOV     2(6),-(6)       ;PUT ADDRESS OF INSTRUCTION ON STACK
016472  162716  000002                  SUB     #2,(6)
016476  011605                          MOV     (6),TTY         ;TYPE (6) IN OCTAL
016500  004767  000212                  JSR     %7,PRINTR       ;TYPE LEADING ZERO'S
016504  000004  017414                  TYPE    ,SPACE+3
016510  010005                          MOV     R0,TTY          ;TYPE R0 IN OCTAL
016512  004767  000200                  JSR     %7,PRINTR       ;TYPE LEADING ZERO'S
016516  000004  017415                  TYPE    ,SPACE+4
016522  012703  001002                  MOV     #ANS1,R3        ;ADDRESS OF DATA
016526  113604                          MOVB    @(6)+,R4        ;AMOUNT OF DATA IN TABLE
016530  001426                          BEQ     3$
016532  100016                          BPL     2$              ;TYPE STACK?
016534  016667  000006  162240          MOV     6(6),ANS1
016542  016667  000010  162234          MOV     10(6),ANS2
016550  016667  000012  162230          MOV     12(6),ANS3
016556  016667  000014  162224          MOV     14(6),ANS4
016564  042704  177600                  BIC     #177600,R4      ;CLEAR SIGN
016570  000004  017415          2$:     TYPE    ,SPACE+4
016574  012305                          MOV     (3)+,TTY        ;TYPE (3)+ IN OCTAL
016576  004767  000114                  JSR     %7,PRINTR       ;TYPE LEADING ZERO'S
016602  005304                          DEC     R4
016604  001371                          BNE     2$
016606  005700                  3$:     TST     FPS
016610  100016                          BPL     4$
016612  000004  017411                  TYPE    ,SPACE
016616  170367  162200                  STST    FEC
016622  016705  162174                  MOV     FEC,TTY         ;TYPE FEC IN OCTAL
016626  004767  000064                  JSR     %7,PRINTR       ;TYPE LEADING ZERO'S
016632  000004  017414                  TYPE    ,SPACE+3
016636  016705  162162                  MOV     FEA,TTY         ;TYPE FEA IN OCTAL
016642  004767  000050                  JSR     %7,PRINTR       ;TYPE LEADING ZERO'S
016646  012604                  4$:     MOV     (6)+,R4
016650  005737  177570                  TST     @#SWR           ;HALT ON ERROR
016654  100001                          BPL     .+4             ;SKIP IF CONTINUE
016656  000000                          HALT                    ;HALT ON ERROR!
016660  032737  001000  177570          BIT     #SW09,@#SWR     ;CHECK FOR INHIBIT LOOP ON ERROR
016666  001001                          BNE     .+4             ;SKIP IF LOOP ON ERROR
016670  000002                          RTI
016672  105067  162103                  CLRB    ICNT+1
016676  032737  000400  177570          BIT     #SW08,@#SWR     ;CHECK FOR LOAD MICROBREAK
016704  001233                          BNE     KIT             ;BRANCH IF NOT
016706  113703  177570                  MOVB    @#SWR,R3        ;PUT MICROBREAK ADDRESS IN R3
016712  170003                          LDCS
016714  000527                          BR      KIT             ;LOOP ON TEST UNTIL NO ERRORS
```

```
016716  012767  000001  000130  PRINTR: MOVB    #1,A4$          ;SET ZERO FILL SWITCH
016724  000402                  BR      .+6
016726  005067  000122          PRINTS: CLR     A4$             ;SUPRESS LEADING ZERO'S
016732  112767  177772  000115          MOVB    #-6.A4$+1       ;SET COUNT
016740  010446                  MOV     R4,-(6)         ;SAVE R4
016742  012704  017044                  MOV     #3$,R4          ;SET POINTER TO FIRST ASCII CHAR.
016746  105014                  CLRB    (4)             ;CLEAR FIRST BYTE
016750  000405                  BR      2$              ;ROTATE FIRST BIT
016752  105014          1$:     CLRB    (4)             ;CLEAR BYTE OF CHARACTER
016754  006105                  ROL     (5)             ;ROTATE BIT INTO C
016756  106114                  ROLB    (4)             ;PACK IT
016760  006105                  ROL     (5)             ;ROTATE BIT INTO C
016762  106114                  ROLB    (4)             ;PACK IT
016764  006105          2$:     ROL     (5)             ;ROTATE BIT INTO C
016766  106114                  ROLB    (4)             ;PACK IT
016770  105714                  TSTB    (4)
016772  001402                  BEQ     .+6
016774  105267  000054          INCB    A4$
017000  105767  000050          TSTB    A4$             ;CHECK FILL SWITCH
017004  001402                  BEQ     .+6
017006  152724  000060          BISB    #'0,(4)+        ;MAKE INTO ASCII CHAR
017012  105267  000037          INCB    A4$+1
017016  001355                  BNE     1$              ;REPEAT
017020  022704  017044                  CMP     #3$,R4
017024  001002                  BNE     .+6
017026  112724  000060                  MOVB    #'0,(4)+
017032  105014                  CLRB    (4)
017034  000004  017044                  TYPE    3$              ;TYPE IT
017040  012604                  MOV     (6)+,R4         ;RESTORE R4
017042  000207                  RTS     PC

017044  000004          3$:     .BLKW   4
017054  000000          A4$:    0

017056  005267  000364  ERROR:  INC     ERRORS          ;COUNT ERRORS
017062  132737  000001  000041          BITB    #1,@#41         ;AUTO MODE?
017070  001412                  BEQ     1$              ;NO!
017072  022767  000010  000346          CMP     #10.,ERRORS     ;TOO MANY?
017100  001006                  BNE     1$              ;NOT YET
017102  013700  000042                  MOV     @#42,R0         ;GET ADDRESS
017106  001403                  BEQ     1$              ;FORGET IT IF ZERO
017110  005037  000042                  CLR     @#42            ;ZAP 42
017114  004710                  JSR     PC,(0)          ;CALL THE MONITOR
017116  000207          1$:     RTS     PC              ;RETURN
```

```
017120  012777  017314  000306  POWDWN: MOV     #ILLUP,@UPVEC   ;SET FOR FAST UP
017126  012777  000340  000302          MOV     #340,@UPVEC+2   ;PRIO:7
017134  170246                          STFPS   -(6)            ;GET THE FPS
017136  170011                          SETD
017140  174046                          STD     AC0,-(6)        ;SAVE AC'S
017142  174146                          STD     AC1,-(6)
017144  174246                          STD     AC2,-(6)
017146  174346                          STD     AC3,-(6)
017150  172404                          LDD     AC4,AC0
017152  174046                          STD     AC0,-(6)
017154  172405                          LDD     AC5,AC0
017156  174046                          STD     AC0,-(6)
017160  010046                          MOV     R0,-(6)         ;SAVE REGISTERS
017162  010146                          MOV     R1,-(6)
017164  010246                          MOV     R2,-(6)
017166  010346                          MOV     R3,-(6)
017170  010446                          MOV     R4,-(6)
017172  010546                          MOV     R5,-(6)
017174  010667  000220                  MOV     SP,SAVE6        ;SAVE SP
017200  012777  017210  000226          MOV     #POWUP,@UPVEC   ;SET UP VECTOR
017206  000000                          HALT

017210  016706  000204          POWUP:  MOV     SAVE6,SP        ;GET SP
017214  005001                          CLR     R1              ;WAIT LOOP FOR THE TTY
017216  005201                  1$:     INC     R1
017220  001376                          BNE     1$
017222  012605                          MOV     (6)+,R5         ;GET THE REGISTERS
017224  012604                          MOV     (6)+,R4
017226  012603                          MOV     (6)+,R3
017230  012602                          MOV     (6)+,R2
017232  012601                          MOV     (6)+,R1
017234  012600                          MOV     (6)+,R0
017236  170011                          SETD
017240  172426                          LDD     (6)+,AC0        ;RESTORE THE AC'S
017242  174005                          STD     AC0,AC5
017244  172426                          LDD     (6)+,AC0
017246  174004                          STD     AC0,AC4
017250  172726                          LDD     (6)+,AC3
017252  172626                          LDD     (6)+,AC2
017254  172526                          LDD     (6)+,AC1
017256  172426                          LDD     (6)+,AC0
017260  170126                          LDFPS   (6)+            ;RESTORE FPS
017262  012777  017120  000140          MOV     #POWDWN,@DWNVEC ;SET UP THE POWER DOWN VECTOR
017270  012777  000340  000134          MOV     #340,@DWNVEC+2
017276  000004  017302                  TYPE    ..+2            ;.ASCIZ <15><12>"POWER"
017302  000002                          RTI

017304  000000                  ILLUP:  HALT                    ;THE POWER UP SEQUENCE WAS STARTED
017306  000777                          BR      .-2             ; BEFORE THE POWER DOWN WAS COMPLETE
```

```
017320  010546                  .IOT:   MOV     IIY,-(6)        ;SAVE IIY
017322  017605  000002                  MOV     2(6),TTY        ;GET ADDRESS TO BE TYPED
017326  105715                  1$:     TSTB    (TTY)           ;TERMINATOR?
017330  001406                          BEQ     2$
017332  112537  177566                  MOVB    (TTY)+,@#177566 ;LOAD AND TYPE THE CHARACTER
017336  105737  177564                  TSTB    @#177564        ;IS THE PRINTER READY
017342  100375                          BPL     .-4
017344  000770                          BR      1$              ;GET THE NEXT CHARACTER
017346  017646  000002          2$:     MOV     2(6),-(6)       ;GET ADDRESS TO BE TYPED
017352  062766  000002  000004          ADD     #2,4(6)         ;ADD 2 TO THE ADDRESS
017360  022666  000002                  CMP     (6)+,2(6        ;IS IT .+2?
017364  001006                          BNE     3$              ;NO
017366  062705  000002                  ADD     #2,TTY          ;ADD 2 TO THE ADDRESS
017372  042705  000001                  BIC     #1,TTY          ;BACK UP TO AN EVEN BYTE
017376  010566  000002                  MOV     TTY,2(6)        ;RESTORE ADDRESS
017402  012605                  3$:     MOV     (6)+,TTY        ;RESTORE TTY
017404  000002                          RTI                     ;RETURN

017406  005015      000         RETURN: .ASCIZ  <15><12>        ;RETURN AND LINEFEED
017411      015  020012  020040 SPACE:  .ASCIZ  <15><12>"   "   ;RETURN AND 3 SPACES
017416      000
        017420                  .EVEN
017420  000000                  SAVE6:  0
017422  172160                  FPTADR: 172160                  ;FLOATING POINT ADDRESS ON THE 11 20
017424  000244  000246          FPVECT: 244,246                 ;FLOATING POINT VECTOR ADDRESS
017430  000024  000026          DWNVEC: 24,26                   ;POWER DOWN VECTOR ADDRESS
017434  000024  000026          UPVEC:  24,26                   ;POWER UP VECTOR ADDRESS
017440  000000                  .TYPE:  0
017442  000000                  TRPB:   0
017444  000000                  LAD:    0                       ;LOOP ADDRESS
017446  000000                  ERRORS: 0                       ;ERROR COUNT
017450  000377                  TIMES:  377                     ;ITERATION COUNT
        000001                  .END
```

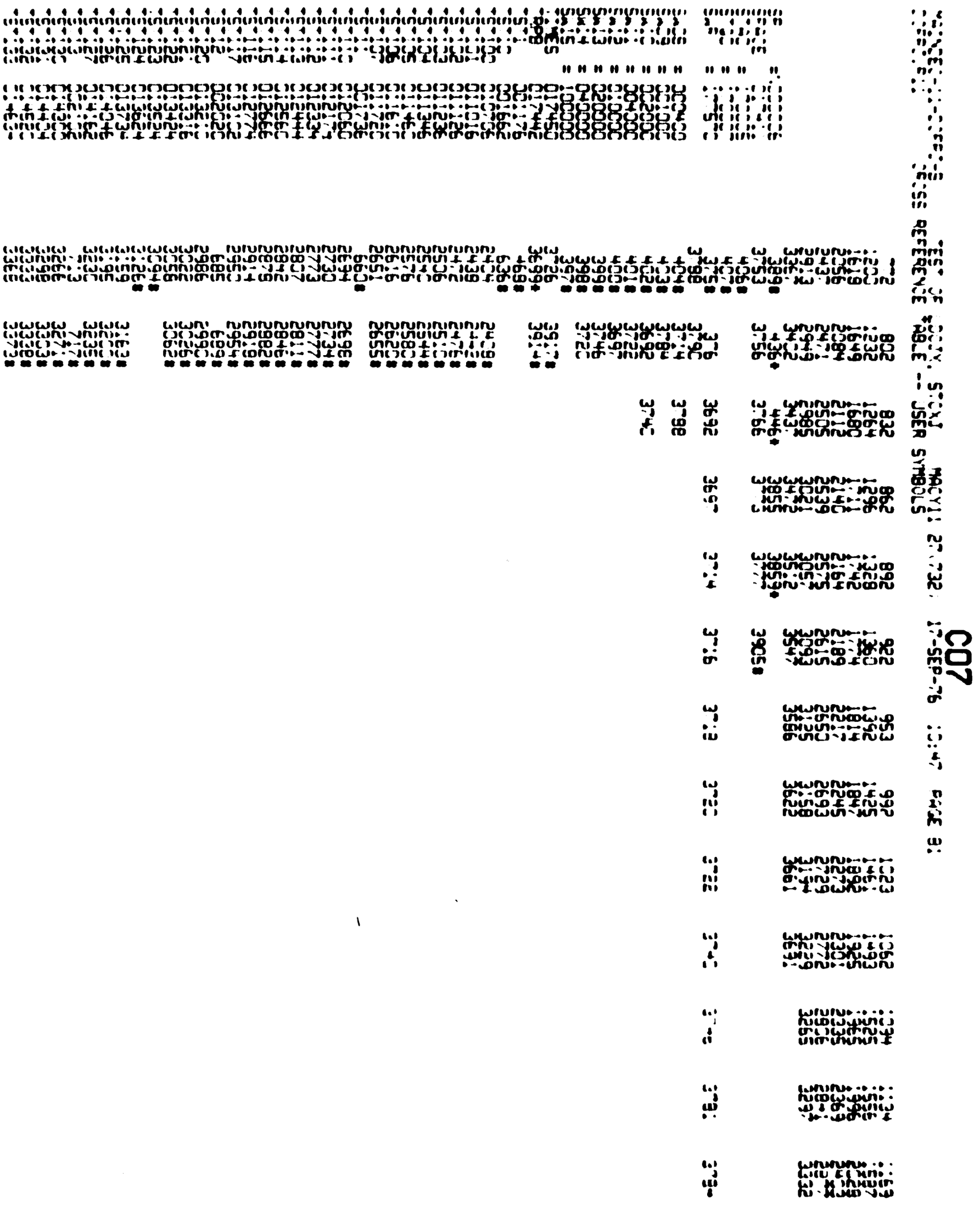
```
AC0     =%000000   391#  517*  523   541*  547   589*  595   718*  719   748*  749   808*  809
                   959*  960   998*  999   1100* 1101  1202* 1208  1234* 1240  1495* 1501  1656*
                  1657   1932* 1933  2034* 2040  2167* 2173  2223* 2229  2279* 2285  2477* 2478
                  2545*  2546  2699* 2701  2883* 2889  3027* 3033  3339* 3340  3444* 3445  3513*
                  3520   3841  3845* 3846  3847* 3848  3870* 3871  3872* 3873  3877*

AC1     =%000001   392#  469*  475   493*  499   662*  668   898*  899   928*  929   1175* 1177
                  1298*  1304  1330* 1336  1362* 1368  1431* 1437  1463* 1469  1625* 1626  1781*
                  1782   1899* 1901  2090* 2096  2341* 2342  2409* 2410  2656* 2657  2847* 2953
                  2991*  2997  3128* 3134  3236* 3237  3409* 3410  3593* 3595  3842  3876*

AC2     =%000002   393#  613*  619   637*  643   690*  696   838*  839   868*  869   1140* 1141
                  1395*  1401  1594* 1595  1749* 1750  1854* 1855  2003* 2005  2142* 2148  2195*
                  2201   2251* 2257  2307* 2308  2443* 2444  2511* 2512  2581* 2582  2621* 2622
                  2778*  2779  2812* 2814  2955* 2961  3095* 3101  3201* 3202  3304* 3305  3479*
                  3480   3843  3875*

AC3     =%000003   394#  565*  571   778*  779   1029* 1030  1068* 1070  1266* 1272  1532* 1533
                  1563*  1564  1687* 1688  1718* 1719  1821* 1822  1972* 1973  2062* 2068  2118*
                  2124   2375* 2376  2735* 2736  2919* 2925  3063* 3069  3164* 3170  3272* 3273
                  3374*  3375  3554* 3555  3629* 3630  3668* 3670  3844  3874*

AC4     =%000004   395#  3845  3873*
AC5     =%000005   396#  3847  3871*
ANR104  011360    2547#  2553
ANR127  014174    3238#  3244
ANR30   003700    1142#  1148
ANR56   006604    1856#  1862
ANS1    001002    425#  475*  476   499*  500   523*  524   547*  548   571*  572   595*  596
                  619*  620   643*  644   668*  669   696*  697   719*  727   749*  757   779*
                   787   809*  817   839*  847   869*  877   899*  907   929*  937   960*  977
                   999*  1007  1030* 1047  1070* 1078  1104* 1119  1148* 1154  1177* 1185  1209*
                  1209   1240* 1241  1272* 1273  1304* 1305  1336* 1337  1368* 1369  1401* 1432
                  1437*  1438  1469* 1470  1501* 1502  1533* 1541  1564* 1572  1595* 1603  1626*
                  1634   1657* 1665  1688* 1696  1719* 1727  1750* 1758  1782* 1799  1925* 1831
                  1862*  1877  1901* 1909  1933* 1950  1973* 1981  2005* 2013  2040* 2041  2068*
                  2069   2096* 2097  2124* 2125  2148* 2149  2173* 2174  2201* 2202  2229* 2230
                  2257*  2258  2285* 2286  2308* 2316  2342* 2350  2376* 2384  2410* 2418  2444*
                  2452   2478* 2486  2512* 2520  2553* 2559  2585* 2600  2622* 2630  2657* 2674
                  2701*  2709  2736* 2753  2779* 2787  2814* 2822  2853* 2854  2889* 2890  2925*
                  2926   2961* 2962  2997* 2998  3033* 3034  3069* 3070  3101* 3102  3134* 3135
                  3170*  3171  3202* 3210  3244* 3250  3276* 3282  3305* 3313  3340* 3348  3375*
                  3383   3410* 3418  3445* 3453  3480* 3493  3520* 3528  3555* 3568  3595* 3603
                  3630*  3643  3670* 3678  3757  3761*
ANS2    001004    426#  480   504   528   552   576   600   624   648   673   701   720*  721*
                   731  750*  751*  761   780*  781*  791   810*  811*  821   840*  841*  851
                   870*  871*  881   900*  901*  911   930*  931*  941   961*  962*  981  1000*
                  1001*  1011  1031* 1032* 1051  1071* 1072* 1082  1102* 1103* 1123  1146* 1147*
                  1158   1178* 1179* 1189  1213  1245  1277  1309  1341  1373  1406  1442  1474
                  1506   1534* 1535* 1545  1565* 1566* 1576  1596* 1597* 1607  1627* 1628* 1638
                  1658*  1659* 1669  1689* 1690* 1700  1720* 1721* 1731  1751* 1752* 1762  1783*
                  1784*  1803  1823* 1824* 1835  1860* 1861* 1881  1902* 1903* 1913  1934* 1935*
                  1954   1974* 1975* 1985  2006* 2007* 2017  2045  2073  2101  2129  2153  2178
                  2206   2234  2262  2290  2320  2354  2388  2422  2456  2490  2524  2551* 2552*
                  2563   2583* 2584* 2604  2634  2678  2713  2757  2791  2826  2858  2894  2930
                  2966   3002  3038  3074  3106  3139  3175  3214  3242* 3243* 3254  3274* 3275*
                  3286   3317  3352  3387  3422  3457  3497  3532  3572  3607  3647  3682  3762*
ANS3    001006    427#  1217  1249  1281  1313  1345  1377  1410  1446  1478  1510  2309* 2310*
                  2324   2343* 2344* 2358  2377* 2378* 2392  2411* 2412* 2426  2445* 2446* 2460
                  2479*  2480* 2494  2513* 2514* 2528  2623* 2624* 2638  2658* 2659* 2682  2703*
```

		2703*	2717	2737*	2738*	2761	2780*	2781*	2795	2815*	2816*	2830	2862	2898
		2934	2970	3006	3042	3078	3110	3147	3179	3203*	3204*	3218	3306*	3307*
		3321	3341*	3342*	3356	3376*	3377*	3391	3411*	3412*	3426	3446*	3447*	3461
		3481*	3482*	3501	3521*	3522*	3536	3556*	3557*	3576	3596*	3597*	3611	3631*
		3632*	3651	3671*	3672*	3686	3763*							
ANS4	001010	428*	1221	1253	1285	1317	1349	1381	1414	1450	1482	1514	2866	2902
		2938	2974	3010	3046	3082	3114	3147	3183	3764*				
ANS5	001012	429*												
ANS6	001014	430*												
ANS7	001016	431*												
ANS8	001020	432*												
ASB	017054	3793*	3795*	3796*	3810*	3811	3814*	3825*						
BEG	001026	415	436*											
BEGIN	001062	440	445*	3702	3704									
BELL =	000207	380*	3694	3742										
DAT100	010702	2406*	2409											
DAT101	011012	2440*	2443											
DAT102	011122	2474*	2477											
DAT103	011232	2508*	2511											
DAT104	011342	2542*	2545											
DAT105	011454	2578*	2581											
DAT106	011604	2618*	2621											
DAT107	011714	2653*	2656											
DAT110	012054	2696*	2699											
DAT111	012170	2732*	2735											
DAT112	012330	2775*	2778											
DAT113	012440	2809*	2812											
DAT114	012554	2844*	2847											
DAT115	012662	2880*	2883											
DAT116	012770	2916*	2919											
DAT117	013076	2952*	2955											
DAT12	002016	687*	690											
DAT120	013204	2988*	2991											
DAT121	013312	3024*	3027											
DAT122	013420	3060*	3063											
DAT125	013730	3161*	3164											
DAT126	014036	3197*	3201											
DAT127	014152	3232*	3236											
DAT13	002076	715*	718											
DAT130	014270	3268*	3272											
DAT131	014374	3300*	3304											
DAT132	014510	3335*	3339											
DAT133	014624	3370*	3374											
DAT134	014740	3405*	3409											
DAT135	015054	3440*	3444											
DAT136	015170	3475*	3479											
DAT137	015322	3515*	3519											
DAT14	002174	745*	748											
DAT140	015436	3550*	3554											
DAT141	015570	3589*	3593											
DAT142	015710	3625*	3629											
DAT143	016042	3664*	3668											
DAT15	002272	775*	778											
DAT16	002370	805*	808											
DAT17	002466	835*	838											
DAT20	002564	865*	868											

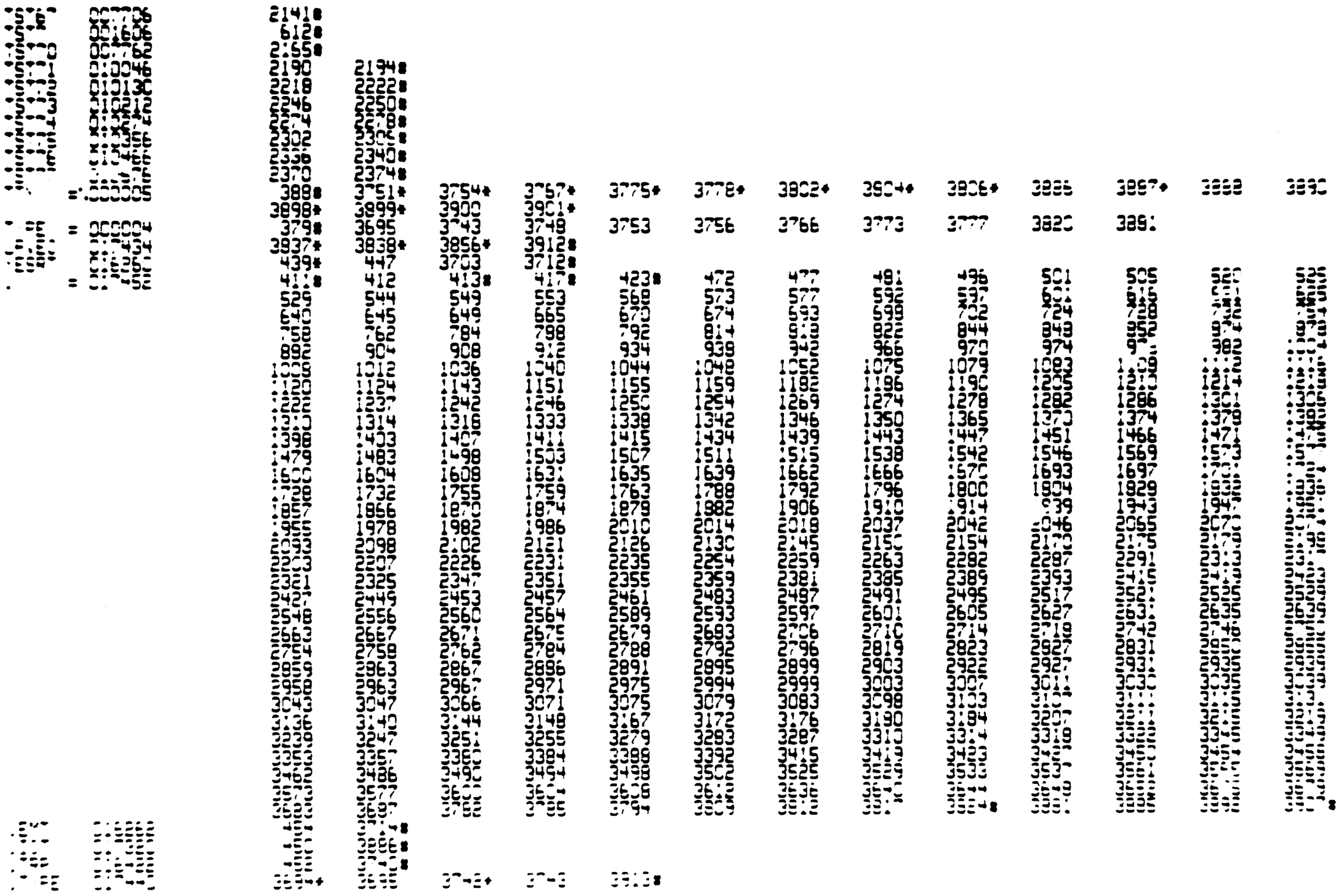
DAT21	002662	895#	898										
DAT22	002760	925#	928										
DAT23	003056	956#	959										
DAT24	003204	995#	998										
DAT25	003302	1026#	1029										
DAT26	003430	1065#	1068										
DAT27	003532	1097#	1100										
DAT30	003662	1137#	1140										
DAT31	003774	1172#	1175										
DAT41	005000	1428#	1431										
DAT44	005304	1528#	1532										
DAT45	005406	1559#	1563										
DAT46	005510	1590#	1594										
DAT47	005612	1621#	1625										
DAT50	005714	1652#	1656										
DAT51	006016	1683#	1687										
DAT52	006120	1714#	1718										
DAT53	006222	1745#	1749										
DAT54	006324	1777#	1781										
DAT55	006456	1817#	1821										
DAT56	006562	1850#	1854										
DAT57	006730	1895#	1899										
DAT60	007036	1928#	1932										
DAT61	007170	1968#	1972										
DAT62	007272	1999#	2003										
DAT63	007400	2031#	2034										
DAT64	007462	2059#	2062										
DAT65	007544	2087#	2090										
DAT66	007626	2115#	2118										
DAT71	010042	2192#	2195										
DAT72	010124	2220#	2223										
DAT73	010206	2248#	2251										
DAT74	010270	2276#	2279										
DAT75	010352	2304#	2307										
DAT76	010462	2338#	2341										
DAT77	010572	2372#	2375										
DISPLA=	177570	376#	3731*	3735*									
DONE	016156	3691#											
DWNVEC	017430	448*	449*	3879*	3880*	3911#							
ERROR	017056	3744	3827#										
ERRORS	017446	3827*	3830	3916#									
FEA	001024	434#	973	1043	1115	1795	1873	1946	2596	2670	2749	3778	
FEC	001022	419*	433#	964*	969	1034*	1039	1106*	1111	1786*	1791	1864*	1869	1937*
		1942	2587*	2592	2661*	2666	2740*	2745	3484*	3489	3559*	3564	3634*	3639
		3774*	3775											
FLTERR	000760	418#	456										
FPI100	010716	2410#											
FPI101	011026	2444#											
FPI102	011136	2478#											
FPI103	011246	2512#											
FPI105	011470	2582#	2596										
FPI106	011620	2622#											
FPI107	011730	2657#	2670										
FPI111	012204	2736#	2749										
FPI112	012344	2779#											
FPI13	002112	719#											

DCFPJ.P11 CROSS REFERENCE TABLE -- USER SYMBOLS

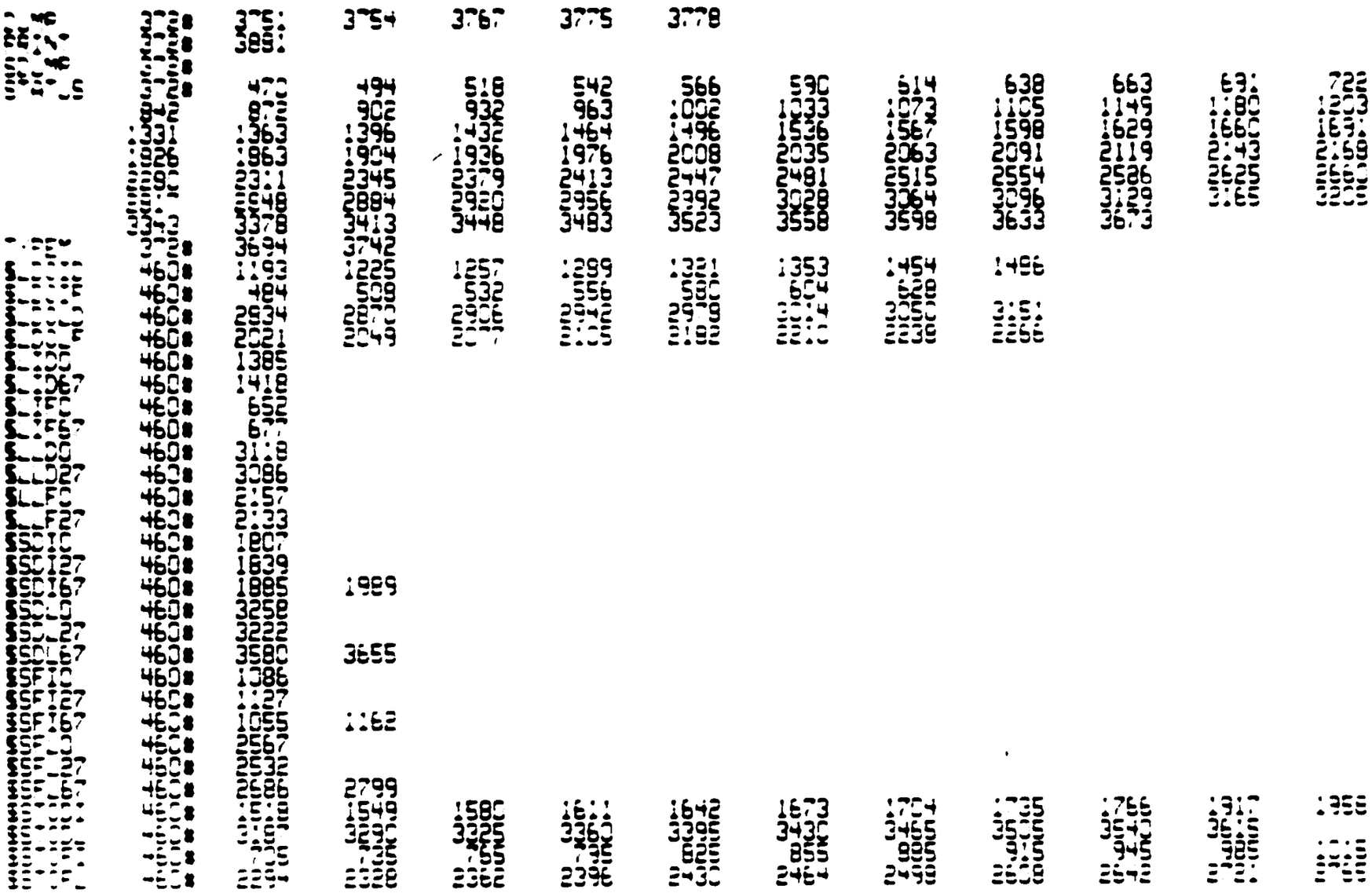
```
FPI14   002210            749#
FPI15   002306            779#
FPI16   002404            809#
FPI17   002502            839#
FPI20   002600            869#
FPI21   002676            899#
FPI22   002774            929#
FPI23   003072            960#    973
FPI24   003220            999#
FPI25   003316           1030#   1043
FPI27   003546           1101#   1115
FPI44   005324           1533#
FPI45   005426           1564#
FPI46   005530           1595#
FPI47   005632           1626#
FPI50   005734           1657#
FPI51   006036           1688#
FPI52   006140           1719#
FPI53   006242           1750#
FPI54   006344           1782#   1795
FPI56   006602           1855#   1873
FPI60   007056           1933#   1946
FPI61   007210           1973#
FPI75   010366           2308#
FPI76   010476           2342#
FPI77   010606           2376#
FPS    =%000000            381#    418#    470#    471    494#    495    518#    519    542#    543    566#    567    590#
                           591     614#    615     638#   639     663#   664     691#   692     722#   723     752#   753
                           782#    783     812#    813    842#    843    872#    873    902#    903    932#    933    963#
                           965    1002#   1003    1033#  1035    1073#  1074    1105#  1107    1149#  1150    1180#  1181
                          1203#   1204    1235#   1236   1267#   1268   1299#   1300   1331#   1332   1363#   1364   1396#
                          1397    1432#   1433    1464#  1465    1496#  1497    1536#  1537    1567#  1568    1598#  1599
                          1629#   1630    1660#   1661   1691#   1692   1722#   1723   1753#   1754   1785#   1787   1826#
                          1827    1863#   1865    1904#  1905    1936#  1938    1976#  1977    2008#  2009    2035#  2036
                          2063#   2064    2091#   2092   2119#   2120   2143#   2144   2168#   2169   2196#   2197   2224#
                          2225    2252#   2253    2280#  2281    2311#  2312    2345#  2346    2379#  2380    2413#  2414
                          2447#   2448    2481#   2482   2515#   2516   2554#   2555   2586#   2588   2625#   2626   2660#
                          2662    2704#   2705    2739#  2741    2782#  2783    2817#  2818    2848#  2849    2884#  2885
                          2920#   2921    2956#   2957   2992#   2993   3026#   3029   3064#   3065   3096#   3097   3129#
                          3130    3165#   3166    3205#  3206    3245#  3246    3277#  3278    3308#  3309    3343#  3344
                          3378#   3379    3413#   3414   3448#   3449   3483#   3485   3523#   3524   3559#   3560   3598#
                          3599    3633#   3635    3673#  3674    3771
FPTADR  017422             442    3909#
FPVECT  017424             456#    457#   3910#
HLT    = 104000            378#    473     478    482    497    502    506    521    526    530    545    550    554
                           569     574     578    593    598    602    617    622    626    641    646    650    666
                           671     675     694    699    703    725    729    733    755    759    763    785    789
                           793     815     819    823    845    849    853    875    879    883    905    909    913
                           935     939     943    967    971    975    979    983   1005   1009   1013   1037   1041
                          1045    1049    1053   1076   1080   1084   1109   1113   1117   1121   1125   1152   1156
                          1160    1183    1187   1191   1206   1211   1215   1219   1223   1238   1243   1247   1251
                          1255    1270    1275   1279   1283   1287   1302   1307   1311   1315   1319   1334   1339
                          1343    1347    1351   1366   1371   1375   1379   1383   1399   1404   1408   1412   1416
                          1435    1440    1444   1448   1452   1467   1472   1476   1480   1484   1499   1504   1508
                          1512    1516    1539   1543   1547   1570   1574   1578   1601   1605   1609   1632   1636
                          1640    1663    1667   1671   1694   1698   1702   1725   1729   1733   1756   1760   1764
```



CROSS REFERENCE TABLE -- USER SYMBOLS



CROSS REFERENCE TABLE -- MACRO NAMES

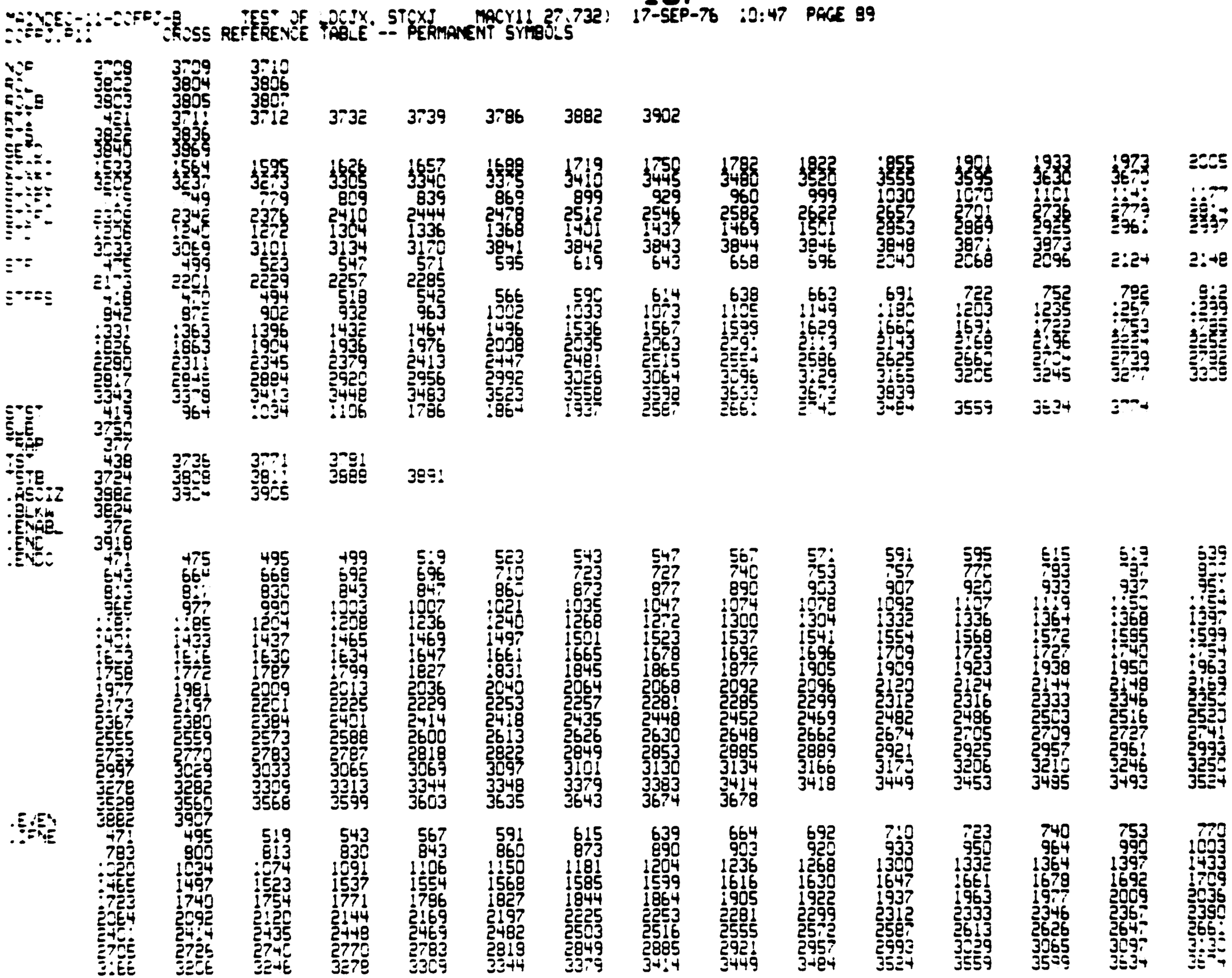


CROSS REFERENCE TABLE -- PERMANENT SYMBOLS

CROSS REFERENCE TABLE -- PERMANENT SYMBOLS

	1572	1576	1599	1603	1607	1630	1634	1638	1661	1665	1669	1692	1696	1720	1723
	1727	1731	1754	1758	1762	1787	1791	1795	1799	1803	1827	1831	1835	1859	1863
	1877	1877	1881	1905	1909	1913	1938	1942	1946	1950	1954	1977	1981	1985	2009
	2013	2017	2036	2041	2045	2064	2069	2073	2092	2097	2101	2120	2125	2129	2144
	2149	2153	2169	2174	2178	2197	2202	2206	2225	2230	2234	2253	2258	2262	2281
	2286	2290	2312	2316	2320	2324	2346	2350	2354	2358	2380	2384	2388	2392	2414
	2419	2423	2426	2448	2452	2456	2460	2482	2486	2490	2494	2516	2520	2524	2528
	2555	2559	2563	2588	2592	2596	2600	2604	2626	2630	2634	2638	2662	2666	2670
	2674	2678	2682	2705	2709	2713	2717	2741	2745	2749	2753	2757	2761	2783	2787
	2791	2795	2818	2822	2826	2830	2849	2854	2958	2862	2866	2885	2890	2894	2998
	2902	2921	2926	2930	2934	2938	2957	2962	2966	2970	2974	2993	2998	3002	3006
	3010	3029	3034	3038	3042	3046	3065	3070	3074	3078	3082	3097	3102	3106	3110
	3114	3130	3135	3139	3143	3147	3166	3171	3175	3179	3183	3206	3210	3214	3218
	3246	3250	3254	3278	3282	3286	3309	3313	3317	3321	3344	3348	3352	3356	3379
	3383	3387	3391	3414	3418	3422	3426	3449	3453	3457	3461	3485	3489	3493	3497
	3501	3524	3528	3532	3536	3560	3564	3568	3572	3576	3599	3603	3607	3611	3635
	3639	3643	3647	3651	3674	3678	3682	3686	3818	3830	3896				
	3716	3720													
	3699														
	3769														
	3780														
	420	1144	1145	1858	1859	2549	2550	3240	3241	3783	3857	3884			
	3882	3886													
	3729	3734	3810	3814											
	3797														
	3707	3744	3752	3755	3768	3776	3779	3835							
	1202	1234	1266	1298	1330	1362	1395	1431	1463	1495					
	469	493	517	541	565	589	613	637	662	690					
	2847	2883	2919	2955	2991	3027	3063	3095	3128	3164					
	2034	2062	2090	2118	2142	2167	2195	2223	2251	2279					
	1532	1563	1594	1625	1656	1687	1718	1749	1781	1821	1854	1899	1932	1972	2003
	3201	3236	3272	3304	3339	3374	3409	3444	3479	3519	3554	3593	3629	3669	3705
	3847	3870	3872	3874	3875	3876	3877								
	718	748	778	808	838	868	898	928	959	998	1029	1068	1100	1140	1175
	2307	2341	2375	2409	2443	2477	2511	2545	2581	2621	2656	2699	2735	2778	2813
	468	492	516	540	564	588	612	636	660	689	717	747	777	805	837
	867	897	927	958	997	1028	1067	1069	1099	1139	1174	1176	1201	1233	1267
	1297	1329	1361	1393	1430	1462	1494	1531	1562	1593	1624	1655	1686	1717	1747
	1780	1820	1853	1898	1900	1931	1971	2002	2004	2033	2061	2089	2111	2141	2175
	1944	2222	2250	2278	2306	2340	2374	2408	2442	2476	2510	2544	2580	2620	2654
	2698	2700	2734	2777	2811	2813	2846	2882	2918	2954	2990	3026	3062	3098	3132
	3163	3200	3235	3271	3303	3338	3373	3408	3443	3478	3518	3553	3592	3594	3628
	3667	3669	3878												
	3719	3791													
	426	437	439	442	445	446	447	448	449	450	451	452	453	454	455
	456	457	661	725	750	780	810	840	870	900	930	961	1000	1031	1061
	1104	1104	1146	1148	1178	1394	1534	1565	1596	1627	1658	1689	1720	1751	1783
	1825	1825	1860	1862	1902	1934	1974	2006	2166	2309	2343	2377	2411	2445	2479
	2551	2551	2553	2583	2585	2623	2658	2702	2737	2780	2815	3121	3203	3237	3271
	3276	3276	3306	3341	3376	3411	3446	3481	3521	3556	3596	3631	3667	3694	3728
	3704	3705	3730	3731	3735	3738	3742	3745	3749	3755	3754	3755	3760	3763	3766
	3767	3767	3776	3778	3780	3797	3798	3821	3832	3837	3838	3849	3850	3853	3854
	3855	3855	3855	3856	3859	3863	3864	3865	3866	3867	3868	3879	3880	3883	3884
	3888	3888	3888												
	3758	3758	3758	3790	3793	3796	3819	3890							

CROSS REFERENCE TABLE -- PERMANENT SYMBOLS



```
.ENC     475   499   523   547   571   595   619   643   668   696   727   757   787   817   847
         903   907   937   969   973  1007  1039  1043  1078  1111  1115  1154  1185  1208  1246
        1252  1304  1336  1368  1401  1437  1469  1501  1541  1572  1603  1634  1665  1696  1727
        1759  1791  1795  1831  1869  1873  1909  1942  1946  1981  2013  2040  2068  2096  2134
        2148  2173  2201  2229  2257  2285  2316  2350  2384  2418  2452  2486  2520  2553  2582
        2596  2630  2666  2670  2709  2745  2749  2787  2822  2853  2889  2925  2961  2997  3033
        3069  3101  3134  3170  3210  3250  3282  3313  3348  3383  3418  3453  3489  3493  3528
        3564  3568  3603  3639  3643  3678
.LIST    372   372   412   422   460   484   508   532   556   580   604   628   652   677   705
         735   765   795   825   855   885   915   945   985  1015  1055  1086  1127  1162  1193
        1225  1257  1289  1321  1353  1385  1418  1454  1486  1518  1549  1580  1611  1642  1673
        1704  1735  1766  1807  1839  1885  1917  1958  1989  2021  2049  2077  2105  2133  2157
        2182  2210  2238  2266  2294  2328  2362  2396  2430  2464  2498  2532  2567  2608  2642
        2686  2721  2765  2799  2834  2870  2906  2942  2978  3014  3050  3086  3119  3151  3187
        3222  3258  3290  3325  3360  3395  3430  3465  3505  3540  3580  3615  3655  3690  3740
        3793  3837  3882  3886
.MACRO   372   460
.MCALL   372
.NLIST   327   372   412   422   460   484   508   532   556   580   604   628   652   677   705
         735   765   795   825   855   885   915   945   985  1015  1055  1086  1127  1162  1193
        1225  1257  1289  1321  1353  1385  1418  1454  1486  1518  1549  1580  1611  1642  1673
        1704  1735  1766  1807  1839  1885  1917  1958  1989  2021  2049  2077  2105  2133  2157
        2182  2210  2238  2266  2294  2328  2362  2396  2430  2464  2498  2532  2567  2608  2642
        2686  2721  2765  2799  2834  2870  2906  2942  2978  3014  3050  3086  3119  3151  3187
        3222  3258  3290  3325  3360  3395  3430  3465  3505  3540  3580  3615  3655  3690  3740
        3793  3837  3882  3886
.PAGE    460  3690
.REM     331
.REPT      2   412
.SBTTL   327   372   422   460  3690  3740  3793  3837  3886
.TITLE   328
```

ERRORS DETECTED: 0
DEFAULT GLOBALS GENERATED: 0

*,DCFPJ.SEQ/SOL/CRF/PAGNUM=DCFPJ
RUN-TIME: 23 35 5 SECONDS
RUN-TIME RATIO: 249/65=3.8
CORE USED: 14K (27 PAGES)

K07


Spooler runtime 12 Seconds, 53 KCS, 351 disk reads, 3 disk writes, 87 pages