

FP11

ADDF ADDD SUBF SUBD
MD-11-DCFPD-C

EP DCFPD-C-DL-A

OCT 1976

COPYRIGHT ©1976

digital

FICHE 1 OF 1

Made in U.S.A.

The microfiche card displays a grid of 48 frames of data, organized into 8 rows and 6 columns. Each frame contains a small table or data set, with some frames showing vertical bar patterns. The data is printed in white on a dark background.

11

MAINDEC-11-DCFPD-C

.REPT 0

IDENTIFICATION

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

COPYRIGHT (C) DIGITAL EQUIPMENT CORPORATION 1973

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

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, LCCDF, STCFD, STCCF
DCFPJ	LDCIF, LDCLF, LDCID, LDCID STCFI, STCFI, STCCI, STCCI
DCFPK	LDEXP, STEXP
DCFPL	MCCF, MCCD

MAINDEC-11-DCFPD-C

11-DCFPD-C

FF11 BASIC INSTRUCTION TEST DCFPA - DCFPL
TABLE OF CONTENTS

CONTENTS

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

11-DCFPD-C

FP11 BASIC INSTRUCTION TEST DCFPA - DCFPL
DESCRIPTION

1. ABSTRACT

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

2. REQUIREMENTS

2.1 EQUIPMENT

PDP11/45 STANDARD COMPUTER WITH FP11 OPTION

2.2 STORAGE

PROGRAM STORAGE - THE ROUTINES USE MEMORY 0 - 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.

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

Vertical mark or signature on the right margin.

E01

MACYDEC-11-20FPC-0
20FPC.P11

TEST OF ADDF, ADDD, SUBF, SUBD MACY11 27(732) 17-SEP-76 09:41 PAGE 4

END

7) THE DISPLAY ON THE 1145 WILL SHOW THE ITERATION COUNT IN
THE LEFT BYTE AND TEST NUMBER IN THE RIGHT. TO USE, SEE THE

FP11 BASIC INSTRUCTION TEST DCFPA - DCFPL
DESCRIPTION

PAGE 4

DATA DISPLAY SWITCH TO THE DISPLAY POSITION.

5. OPERATING PROCEDURE

5.1 OPERATIONAL SWITCH SETTINGS

AT SA 200 ... ALL SWITCHES DOWN IS WORST CASE TESTING. 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 JB 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 LOOPEL 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 LOOPEL UPON UNTIL 256 CONSECUTIVE GOOD PASSES ARE COMPLETED IF SW<9> IS ON A 1. TO INHIBIT TYPEOUTS, PUT SW<13> ON A 1.

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

FP11 BASIC INSTRUCTION TEST DCFPA - DCFPL
DESCRIPTION

PAGE 6

6.2 ERROR RECOVERY
RESTART AT 200

7. RESTRICTIONS
NONE

8. MISCELLANEOUS

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

8.2 STACK POINTER
STACK IS INITIALLY SET TO 600

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

9. PROGRAM DESCRIPTION

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

FP11 BASIC INSTRUCTION TEST DCFPA - DCFPL
DESCRIPTION
PAGE 6

.TITLE MAINDEC-11-DCFPD-C TEST OF ADDF, ADDD, SUBF, SUBD
:COPYRIGHT 1972, 1973 DIGITAL EQUIPMENT CORP., MAYNARD, MASS
:PROGRAM BY KEN CHAPMAN
.REM*

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

OUTPUT FORM:

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

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

```

000001      .ENABL  ABS
177776      N=      1
177570      PS=     177776
177570      SWR=     177570
177570      DISPLAY=SWR
104400      SCOPE=   TRAP
104000      HLT=     EMT
000004      TYPE=    IOT
000207      BELL=    207
000000      FPS=     %0
000000      R0=      %0
000001      R1=      %1
000002      R2=      %2
000003      R3=      %3
000204      R4=      %4
000005      R5=      %5
000005      TTY=     %5
000006      SP=      %6
000007      PC=      %7
000000      ACC=     %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      STAQ=    170005
170007      STQ0=    170007
170006      MRS=    170006
170004      LDSC=    170004

000000      .=      0
000200      .=      200

000200 000167 000622      JMP      BEG

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

```

:TRAP CATCHER FROM 0 - 776

```

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

001026 012706 000600      BEG:  MOV    #600,SP          ; ** STACK AT 600 **
001032 012737 001054 000004      MOV    #M1120,2#4      ; FIND OUT WHICH MACHINE THIS IS
001040 005737 177772          TST    2#177772        ; IS PIRQ THERE?
001044 012767 000006 015426      MOV    #6,YESRT      ; FUDGE IN RTT IF 11/45
001052 000403          BR      BEGIN

001054 016737 016562 000010      M1120: MOV   FPTADR,2#10      ; LOAD THE ILLEGAL INSTRUCTION VECTOR
                                ; WITH THE ADDRESS OF THE FPJ.
                                ; THE FPU WILL HANDLE THE BAD OPCODES

001062 012737 000006 000004      BEGIN: MOV   #6,2#4          ; RESET 4
001070 012706 000600          MOV   #600,SP
001074 012737 016500 000014      MOV   #YESRT,2#14      ; SET TRACE TRAP VECTOR
001102 012777 017340 016540      MOV   #POWDWN,2DWNVEC
001110 012777 000340 016534      MOV   #340,2DWNVEC+2
001116 012737 017540 000020      MOV   #.IOT,2#20       ; SET UP VECTOR 20
001124 012700 000030          MOV   #30,R0           ; SET R0 TO VECTOR 30
001130 012720 016642          MOV   #.TRP,(0)+       ; SET EMT VECTOR
001134 012720 000340          MOV   #340,(0)+
001140 012720 016502          MOV   #.EMT,(0)+      ; SET TRAP VECTOR
001144 012710 000340          MOV   #340,(0)
001150 012777 000760 016466      MOV   #FLTERR,2FPVECT  ; LOAD INTERRUPT VECTOR
001156 012777 000340 016462      MOV   #340,2FPVECT+2  ; LOCK UP PROCESSOR
001164 005067 177610          CLR   ICNT
001170 005067 016470          CLR   LAC

```

```

*****
:TEST 1: TEST ADDF (ADD FLOATING)
:      000004,005000 + 000030,000200 = 000000,000000
:      FPS = 047404, FSRC = M6-R7, AC = AC3
*****

```

```

001174 104400          SCOPE
001176 000404          BR      TST1          ;BRANCH OVER INPUT DATA

001200 000004 005000  DTA1: 000004,005000
001204 000030 000200  DTB1: 0000030,000200

001210 170127 047417  TST1: LDFPS #047417          ;LOAD FLOATING POINT STATUS
001214 172767 177760  LDF      DTA1, AC3          ;LOAD 000004,005000 INTO AC3
001220 172367 177760  FP11: ADDF      DTB1, AC3          ;ADD 0000030,000200 TO AC3
001224 170200          STFPS      FPS              ;STORE FLOATING POINT STATUS
001226 022700 047404  CMP      #047404,FPS        ;CHECK FLOATING POINT STATUS
001232 001401          BEQ      .+4              ;BRANCH IF OK
001234 104000          HLT              ;FPS NOT EQUAL TO 047404

001236 174367 177540  STF      AC3, ANS1          ;STORE SUM IN ANS1, ANS2
001242 022767 000000 177532  CMP      #000000,ANS1        ;CHECK ANS1
001250 001401          BEQ      .+4              ;BRANCH IF OK
001252 104002          HLT+2          ;ANS1 NOT EQUAL TO 000000

001254 022767 000000 177522  CMP      #000000,ANS2        ;CHECK ANS2
001262 001401          BEQ      .+4              ;BRANCH IF OK
001264 104002          HLT+2          ;ANS2 NOT EQUAL TO 000000

```

```

*****
:TEST 2: TEST ADDF (ADD FLOATING)
:      040125,052525 + 140052,125252 = 037452,125254
:      FPS = 047400, FSRC = M6-R7, AC = AC2
*****

```

```

001266 104400          SCOPE
001270 000404          BR      TST2          ;BRANCH OVER INPUT DATA

001272 040125 052525  DTA2: 040125,052525
001276 140052 125252  DTB2: 140052,125252

001302 170127 047417  TST2: LDFPS #047417          ;LOAD FLOATING POINT STATUS
001306 172667 177760  LDF      DTA2, AC2          ;LOAD 040125,052525 INTO AC2
001312 172267 177760  FP12: ADDF      DTB2, AC2          ;ADD 140052,125252 TO AC2
001316 170200          STFPS      FPS              ;STORE FLOATING POINT STATUS
001320 022700 047400  CMP      #047400,FPS        ;CHECK FLOATING POINT STATUS
001324 001401          BEQ      .+4              ;BRANCH IF OK
001326 104000          HLT              ;FPS NOT EQUAL TO 047400

001330 174267 177446  STF      AC2, ANS1          ;STORE SUM IN ANS1, ANS2
001334 022767 037452 177440  CMP      #037452,ANS1        ;CHECK ANS1
001342 001401          BEQ      .+4              ;BRANCH IF OK
001344 104002          HLT+2          ;ANS1 NOT EQUAL TO 037452

```

MO1

MAINDEC-11-DCFPD-C
DCFPD.P11

TEST OF ADDF, ADDD, SUBF, SUBD MACY11 27(732) 17-SEP-76 09:41 PAGE 12
TEST SECTION

001346 022767 125254 177430 CMP #125254,ANS2 ;CHECK ANS2
001354 001401 BEQ .+4 ;BRANCH IF OK
001356 104002 HLT+2 ;ANS2 NOT EQUAL TO 125254

:TEST 3: TEST ADDF (ADD FLOATING)
: 040052,125252 + 040052,125252 = 040252,125252
: FPS = 047400, FSRC = M6-R7, AC = AC1
:*****

001360 104400 SCOPE
001362 000404 BR TST3 ;BRANCH OVER INPUT DATA

001364 040052 125252 DTA3: 040052,125252
001370 040052 125252 DTB3: 040052,125252

001374 170127 047417 TST3: LDFPS #047417 ;LOAD FLOATING POINT STATUS
001400 172567 177760 LDF DTA3, AC1 ;LOAD 040052,125252 INTO AC1
001404 172167 177760 FPI3: ADDF DTB3, AC1 ;ADD 040052,125252 TO AC1
001410 170200 STFPS FPS ;STORE FLOATING POINT STATUS
001412 022700 047400 CMP #047400,FPS ;CHECK FLOATING POINT STATUS
001416 001401 BEQ .+4 ;BRANCH IF OK
001420 104000 HLT ;FPS NOT EQUAL TO 047400

001422 174167 177354 STF AC1, ANS1 ;STORE SUM IN ANS1, ANS2
001426 022767 040252 177346 CMP #040252,ANS1 ;CHECK ANS1
001434 001401 BEQ .+4 ;BRANCH IF OK
001436 104002 HLT+2 ;ANS1 NOT EQUAL TO 040252

001440 022767 125252 177336 CMP #125252,ANS2 ;CHECK ANS2
001446 001401 BEQ .+4 ;BRANCH IF OK
001450 104002 HLT+2 ;ANS2 NOT EQUAL TO 125252

:TEST 4: TEST ADDF (ADD FLOATING)
: 077777,177777 + 177777,177777 = 000000,000000
: FPS = 047404, FSRC = M6-R7, AC = AC3
:*****

001452 104400 SCOPE
001454 000404 BR TST4 ;BRANCH OVER INPUT DATA

001456 077777 177777 DTA4: 077777,177777
001462 177777 177777 DTB4: 177777,177777

001466 170127 047417 TST4: LDFPS #047417 ;LOAD FLOATING POINT STATUS
001472 172767 177760 LDF DTA4, AC3 ;LOAD 077777,177777 INTO AC3
001476 172367 177760 FPI4: ADDF DTB4, AC3 ;ADD 177777,177777 TO AC3
001502 170200 STFPS FPS ;STORE FLOATING POINT STATUS
001504 022700 047404 CMP #047404,FPS ;CHECK FLOATING POINT STATUS
001510 001401 BEQ .+4 ;BRANCH IF OK
001512 104000 HLT ;FPS NOT EQUAL TO 047404

001514 174367 177262 STF AC3, ANS1 ;STORE SUM IN ANS1, ANS2

```

001520 022767 000000 177254      CMP      #000000,ANS1      ;CHECK ANS1
001526 001401      BEQ      .+4              ;BRANCH IF OK
001530 104002      HLT+2          ;ANS1 NOT EQUAL TO 000000

001532 022767 000000 177244      CMP      #000000,ANS2      ;CHECK ANS2
001540 001401      BEQ      .+4              ;BRANCH IF OK
001542 104002      HLT+2          ;ANS2 NOT EQUAL TO 000000

```

```

;*****
;TEST 5:      TEST ADDF (ADD FLOATING)
;             040252,125252 + 037525,052525 = 040305,052525
;             FPS = 047400,   FSRC = M6-R7,   AC = AC2
;*****

```

```

001544 104400      SCOPE
001546 000404      BR      TST5          ;BRANCH OVER INPUT DATA

001550 040252 125252      DTA5:   040252,125252
001554 037525 052525      DTB5:   037525,052525

001560 170127 047417      TST5:   LDFPS      #047417      ;LOAD FLOATING POINT STATUS
001564 172667 177760      LDF      DTA5,   AC2      ;LOAD 040252,125252 INTO AC2
001570 172267 177760      FPI5:   ADDF      DTB5,   AC2      ;ADD 037525,052525 TO AC2
001574 170200      STFPS      FPS          ;STORE FLOATING POINT STATUS
001576 022700 047400      CMP      #047400,FPS      ;CHECK FLOATING POINT STATUS
001602 001401      BEQ      .+4              ;BRANCH IF OK
001604 104000      HLT          ;FPS NOT EQUAL TO 047400

001606 174267 177170      STF      AC2,   ANS1      ;STORE SUM IN ANS1, ANS2
001612 022767 040305 177162      CMP      #040305,ANS1      ;CHECK ANS1
001620 001401      BEQ      .+4              ;BRANCH IF OK
001622 104002      HLT+2          ;ANS1 NOT EQUAL TO 040305

001624 022767 052525 177152      CMP      #052525,ANS2      ;CHECK ANS2
001632 001401      BEQ      .+4              ;BRANCH IF OK
001634 104002      HLT+2          ;ANS2 NOT EQUAL TO 052525

```

```

;*****
;TEST 6:      TEST ADDF (ADD FLOATING)
;             137525,052525 + 140253 = 140305,125253
;             FPS = 047410,   FSRC = M2-R7,   AC = AC1
;*****

```

```

001636 104400      SCOPE
001640 000402      BR      TST6          ;BRANCH OVER INPUT DATA

001642 137525 052525      DTA6:   137525,052525

001646 170127 047417      TST6:   LDFPS      #047417      ;LOAD FLOATING POINT STATUS
001652 172567 177764      LDF      DTA6,   AC1      ;LOAD 137525,052525 INTO AC1
001656 172127 140253      FPI6:   ADDF      #-1.333,AC1      ;ADD 140253 TO AC1
001662 170200      STFPS      FPS          ;STORE FLOATING POINT STATUS
001664 022700 047410      CMP      #047410,FPS      ;CHECK FLOATING POINT STATUS
001670 001401      BEQ      .+4              ;BRANCH IF OK

```

```

001672 104000          HLT          :FPS NOT EQUAL TO 047410
001674 174167 177102   STF          AC1          ANS1      :STORE SUM IN ANS1, ANS2
001700 022767 140305 177074  CMP          #140305,ANS1      :CHECK ANS1
001706 001401       BEQ          .+4           :BRANCH IF OK
001710 104002       HLT+2          :ANS1 NOT EQUAL TO 140305

001712 022767 125253 177064  CMP          #125253,ANS2      :CHECK ANS2
001714 001401       BEQ          .+4           :BRANCH IF OK
001716 104002       HLT+2          :ANS2 NOT EQUAL TO 125253

```

```

*****
:TEST 7:          TEST ADDF (ADD FLOATING)
:                077777,177777 + 000200,000000 = 077777,177777
:                FPS = 047400,  FSRC = M6-R7,  AC = AC3
*****

```

```

001724 104400          SCOPE
001726 000404          BR          *5*7          :BRANCH OVER INPUT DATA

001730 077777 177777   DTA7:       077777,177777
001734 000200 000000  DTB7:       000200,000000

001740 170127 047417   *5*7:       LDFPS        #047417      :LOAD FLOATING POINT STATUS
001744 172767 177760   LDF          DTA7,  AC3      :LOAD 077777,177777 INTO AC3
001750 172367 177760   FP17:       ADDF         DTB7,  AC3      :ADD 000200,000000 TO AC3
001754 170200       STFPS        FPS          :STORE FLOATING POINT STATUS
001756 022700 047400   CMP          #047400,FPS      :CHECK FLOATING POINT STATUS
001762 001401       BEQ          .+4           :BRANCH IF OK
001764 104000       HLT          :FPS NOT EQUAL TO 047400

001766 174367 177010   STF          AC3          ANS1      :STORE SUM IN ANS1, ANS2
001772 022767 077777 177002  CMP          #077777,ANS1      :CHECK ANS1
002000 001401       BEQ          .+4           :BRANCH IF OK
002002 104002       HLT+2          :ANS1 NOT EQUAL TO 077777

002004 022767 177777 176772  CMP          #177777,ANS2      :CHECK ANS2
002012 001401       BEQ          .+4           :BRANCH IF OK
002014 104002       HLT+2          :ANS2 NOT EQUAL TO 177777

```

```

*****
:TEST 10:         TEST ADDF (ADD FLOATING)
:                000200,000000 + 177777,177777 = 177777,177777
:                FPS = 047410,  FSRC = M6-R7,  AC = AC2
*****

```

```

002016 104400          SCOPE
002020 000404          BR          *5*10         :BRANCH OVER INPUT DATA

002022 000200 000000   DTA10:      000200,000000
002026 177777 177777   DTB10:      177777,177777

002032 170127 047417   *5*10:     LDFPS        #047417      :LOAD FLOATING POINT STATUS
002036 172367 177760   LDF          DTA10, AC2      :LOAD 000200,000000 INTO AC2

```

002042	172267	177760		FPI10:	ADDF	DTB10, AC2	:ADD 177777,177777 TO AC2
002046	170200				STFPS	FPS	:STORE FLOATING POINT STATUS
002050	022700	047410			CMP	#047410,FPS	:CHECK FLOATING POINT STATUS
002054	001401				BEG	.+4	:BRANCH IF OK
002056	104000				HLT		:FPS NOT EQUAL TO 047410
002060	174267	176716			STF	AC2, ANS1	:STORE SUM IN ANS1, ANS2
002064	022767	177777	176710		CMP	#177777,ANS1	:CHECK ANS1
002070	001401				BEG	.+4	:BRANCH IF OK
002074	104002				HLT+2		:ANS1 NOT EQUAL TO 177777
002076	022767	177777	176700		CMP	#177777,ANS2	:CHECK ANS2
002104	001401				BEG	.+4	:BRANCH IF OK
002106	104002				HLT+2		:ANS2 NOT EQUAL TO 177777

TEST 11: TEST ADDF (ADD FLOATING)
035152,125252 + 043125,052525 = 043125,052526
FPS = 047400, FSRC = M6-R7, AC = AC3

002110	104400				SCOPE		
002112	000404				BR	TST11	:BRANCH OVER INPUT DATA
002114	035152	125252		DTA11:	035152,125252		
002120	043125	052525		DTB11:	043125,052525		
002124	170127	047417		TST11:	LDFPS	#047417	:LOAD FLOATING POINT STATUS
002130	172767	177760			LDF	DTA11, AC3	:LOAD 035152,125252 INTO AC3
002134	172367	177760		FPI11:	ADDF	DTB11, AC3	:ADD 043125,052525 TO AC3
002140	170200				STFPS	FPS	:STORE FLOATING POINT STATUS
002142	022700	047400			CMP	#047400,FPS	:CHECK FLOATING POINT STATUS
002146	001401				BEG	.+4	:BRANCH IF OK
002150	104000				HLT		:FPS NOT EQUAL TO 047400
002152	174367	176624			STF	AC3, ANS1	:STORE SUM IN ANS1, ANS2
002156	022767	043125	176616		CMP	#043125,ANS1	:CHECK ANS1
002164	001401				BEG	.+4	:BRANCH IF OK
002166	104002				HLT+2		:ANS1 NOT EQUAL TO 043125
002170	022767	052526	176606		CMP	#052526,ANS2	:CHECK ANS2
002176	001401				BEG	.+4	:BRANCH IF OK
002200	104002				HLT+2		:ANS2 NOT EQUAL TO 052526

TEST 12: TEST ADDF (ADD FLOATING)
035152,125252 + 043325 = 043325,000000
FPS = 047400, FSRC = M2-R7, AC = AC1

002202	104400				SCOPE		
002204	000402				BR	TST12	:BRANCH OVER INPUT DATA
002206	035152	125252		DTA12:	035152,125252		


```

002212 170127 047417 TST12: LDFPS #047417 :LOAD FLOATING POINT STATUS
002216 172567 177764 LDF DTB12, AC1 :LOAD 035152,125252 INTO AC1
002222 172127 043325 FPI12: ADDF #6916,AC1 :ADD 043325 TO AC1
002226 170200 STFPS FPS :STORE FLOATING POINT STATUS
002230 022700 047400 CMP #047400,FPS :CHECK FLOATING POINT STATUS
002234 001401 BEQ .+4 :BRANCH IF OK
002236 104000 HLT :FPS NOT EQUAL TO 047400

002240 174167 176536 STF AC1,ANS1 :STORE SUM IN ANS1, ANS2
002244 022767 043325 176530 CMP #043325,ANS1 :CHECK ANS1
002252 001401 BEQ .+4 :BRANCH IF OK
002254 104002 HLT+2 :ANS1 NOT EQUAL TO 043325

002256 022767 000000 176520 CMP #000000,ANS2 :CHECK ANS2
002264 001401 BEQ .+4 :BRANCH IF OK
002266 104002 HLT+2 :ANS2 NOT EQUAL TO 000000

```

```

*****
:TEST 13: TEST ADDF (ADD FLOATING)
: 035152,125252 + 143125,052525 = 143125,052524
: FPS = 047410, FSRC = M6-R7, AC = ACC
*****

```

```

002270 104400 SCOPE
002272 000404 BR TST13 :BRANCH OVER INPUT DATA

002274 035152 125252 DTB13: 035152,125252
002300 143125 052525 DTB13: 143125,052525

002304 170127 047417 TST13: LDFPS #047417 :LOAD FLOATING POINT STATUS
002310 172467 177760 LDF DTB13, ACC :LOAD 035152,125252 INTO ACC
002314 172057 177760 FPI13: ADDF DTB13, ACC :ADD 143125,052525 TO ACC
002320 170200 STFPS FPS :STORE FLOATING POINT STATUS
002322 022700 047410 CMP #047410,FPS :CHECK FLOATING POINT STATUS
002326 001401 BEQ .+4 :BRANCH IF OK
002330 104000 HLT :FPS NOT EQUAL TO 047410

002332 174067 176444 STF ACC,ANS1 :STORE SUM IN ANS1, ANS2
002336 022767 143125 176426 CMP #143125,ANS1 :CHECK ANS1
002344 001401 BEQ .+4 :BRANCH IF OK
002346 104002 HLT+2 :ANS1 NOT EQUAL TO 143125

002350 022767 052524 176426 CMP #052524,ANS2 :CHECK ANS2
002356 001401 BEQ .+4 :BRANCH IF OK
002360 104002 HLT+2 :ANS2 NOT EQUAL TO 052524

```

```

*****
:TEST 14: TEST ADDF (ADD FLOATING)
: 035152,125252 + 142725,052525 = 142725,052523
: FPS = 047410, FSRC = M6-R7, AC = ACC
*****

```

```

002362 104000 SCOPE

```

```

002364 000404 BR TST14 :BRANCH OVER INPLT DATA
002366 035152 125252 DTA14: 035152,125252
002372 142725 052525 DTB14: 142725,052525
002376 170127 047417 TST14: LDFPS #047417 :LOAD FLOATING POINT STATUS
002402 172467 177760 LDF DTA14, ACC :LOAD 035152,125252 INTO ACC
002406 172067 177760 FP114: ADDF DTB14, ACC :ADD 142725,052525 TO ACC
002410 170200 STFPS FPS :STORE FLOATING POINT STATUS
002414 022700 047410 CMP #047410,FPS :CHECK FLOATING POINT STATUS
002420 001401 BEQ .+4 :BRANCH IF OK
002422 104000 HLT :FPS NOT EQUAL TO 047410
002424 174067 176352 STF ACC, ANS1 :STORE SUM IN ANS1, ANS2
002430 022767 142725 176344 CMP #142725,ANS1 :CHECK ANS1
002436 001401 BEQ .+4 :BRANCH IF OK
002440 104002 HLT+2 :ANS1 NOT EQUAL TO 142725
002442 022767 052523 176334 CMP #052523,ANS2 :CHECK ANS2
002450 001401 BEQ .+4 :BRANCH IF OK
002452 104002 HLT+2 :ANS2 NOT EQUAL TO 052523

```

```

*****
:TEST 15: TEST ADDF (ADD FLOATING)
:077652,125252 + 077452,125252 = 077777,177777
:FPS = 047400, FSRC = M6-R7, AC = AC:
*****

```

```

002454 104400 SCOPE
002456 000404 BR TST15 :BRANCH OVER INPLT DATA
002460 077652 125252 DTA15: 077652,125252
002464 077452 125252 DTB15: 077452,125252
002470 170127 047417 TST15: LDFPS #047417 :LOAD FLOATING POINT STATUS
002474 172567 177760 LDF DTA15, AC1 :LOAD 077652,125252 INTO AC1
002500 172167 177760 FP15: ADDF DTB15, AC1 :ADD 077452,125252 TO AC1
002504 170200 STFPS FPS :STORE FLOATING POINT STATUS
002506 022700 047400 CMP #047400,FPS :CHECK FLOATING POINT STATUS
002512 001401 BEQ .+4 :BRANCH IF OK
002514 104000 HLT :FPS NOT EQUAL TO 047400
002516 174167 176260 STF AC1, ANS1 :STORE SUM IN ANS1, ANS2
002522 022767 077777 176252 CMP #077777,ANS1 :CHECK ANS1
002530 001401 BEQ .+4 :BRANCH IF OK
002532 104002 HLT+2 :ANS1 NOT EQUAL TO 077777
002534 022767 177777 176242 CMP #177777,ANS2 :CHECK ANS2
002540 001401 BEQ .+4 :BRANCH IF OK
002544 104002 HLT+2 :ANS2 NOT EQUAL TO 177777

```

```

*****
:TEST 16: TEST ADDF (ADD FLOATING)
:177452,125252 + 177652,125252 = 177777,177777
*****

```

: FPS = 047410, FSRC = M6-R7, AC = AC3
:*****

002546	104400			SCOPE		
002550	000404			BR	TST16	:BRANCH OVER INPUT DATA
002552	177452	125252		DTA16:	177452,125252	
002556	177652	125252		DTB16:	177652,125252	
002562	170127	047417		TST16:	LDFPS #047417	:LOAD FLOATING POINT STATUS
002566	172767	177760			LDF DTA16, AC3	:LOAD 177452,125252 INTO AC3
002572	172367	177760		FPI16:	ADDF DTB16, AC3	:ADD 177652,125252 TO AC3
002576	170200				STFPS FPS	:STORE FLOATING POINT STATUS
002600	022700	047410			CMP #047410,FPS	:CHECK FLOATING POINT STATUS
002604	001401				BEQ .+4	:BRANCH IF OK
002606	104000				HLT	:FPS NOT EQUAL TO 047410
002610	174367	176166		STF	AC3, ANS1	:STORE SUM IN ANS1, ANS2
002614	022767	177777	176160	CMP	#177777,ANS1	:CHECK ANS1
002622	001401			BEQ	.+4	:BRANCH IF OK
002624	104002			HLT+2		:ANS1 NOT EQUAL TO 177777
002626	022767	177777	176160	CMP	#177777,ANS2	:CHECK ANS2
002634	001401			BEQ	.+4	:BRANCH IF OK
002636	104002			HLT+2		:ANS2 NOT EQUAL TO 177777

:*****
:TEST 17: TEST ADDF (ADD FLOATING)
: 000425,052525 + 100252,125252 = 000200,000000
: FPS = 047400, FSRC = M6-R7, AC = AC3
:*****

002640	104400			SCOPE		
002642	000404			BR	TST17	:BRANCH OVER INPUT DATA
002644	000425	052525		DTA17:	000425,052525	
002650	100252	125252		DTB17:	100252,125252	
002654	170127	047417		TST17:	LDFPS #047417	:LOAD FLOATING POINT STATUS
002660	172767	177760			LDF DTA17, AC3	:LOAD 000425,052525 INTO AC3
002664	172367	177760		FPI17:	ADDF DTB17, AC3	:ADD 100252,125252 TO AC3
002670	170200				STFPS FPS	:STORE FLOATING POINT STATUS
002672	022700	047400			CMP #047400,FPS	:CHECK FLOATING POINT STATUS
002676	001401				BEQ .+4	:BRANCH IF OK
002700	104000				HLT	:FPS NOT EQUAL TO 047400
002702	174367	176074		STF	AC3, ANS1	:STORE SUM IN ANS1, ANS2
002706	022767	000200	176066	CMP	#000200,ANS1	:CHECK ANS1
002714	001401			BEQ	.+4	:BRANCH IF OK
002716	104002			HLT+2		:ANS1 NOT EQUAL TO 000200
002720	022767	000000	176066	CMP	#000000,ANS2	:CHECK ANS2
002726	001401			BEQ	.+4	:BRANCH IF OK
002730	104002			HLT+2		:ANS2 NOT EQUAL TO 000000

```

*****
:TEST 20:           TEST ADDF (ADD FLOATING)
:           100252,125252 + 000425,052525 = 000200,000000
:           FPS = 047400,   FSRC = M6-R7,   AC = AC2
*****
  
```

002732	104400			SCOPE		
002734	000404			BR	TST20	;BRANCH OVER INPLT DATA
002736	100252	125252		DTA20:	100252,125252	
002742	000425	052525		DTB20:	000425,052525	
002746	170127	047417		TST20:	LDFPS	#047417 ;LOAD FLOATING POINT STATUS
002752	172667	177760			LDF	DTA20, AC2 ;LOAD 100252,125252 INTO AC2
002756	172267	177760		FPI20:	ADDF	DTB20, AC2 ;ADD 000425,052525 TO AC2
002762	170200				STFPS	FPS ;STORE FLOATING POINT STATUS
002764	022700	047400			CMP	#047400,FPS ;CHECK FLOATING POINT STATUS
002770	001401				BEG	.+4 ;BRANCH IF OK
002772	104000				HLT	;FPS NOT EQUAL TO 047400
002774	174267	176002		STF	AC2, ANS1	;STORE SUM IN ANS1, ANS2
003000	022767	000200	175774	CMP	#000200,ANS1	;CHECK ANS1
003006	001401			BEG	.+4	;BRANCH IF OK
003010	104002			HLT+2		;ANS1 NOT EQUAL TO 000200
003012	022767	000000	175764	CMP	#000000,ANS2	;CHECK ANS2
003020	001401			BEG	.+4	;BRANCH IF OK
003022	104002			HLT+2		;ANS2 NOT EQUAL TO 000000

```

*****
:TEST 21:           TEST ADDF (ADD FLOATING)
:           100177,177777 + 101252,125252 = 101252,125252
:           FPS = 047410,   FSRC = M6-R7,   AC = AC2
*****
  
```

003024	104400			SCOPE		
003026	000404			BR	TST21	;BRANCH OVER INPUT DATA
003030	100177	177777		DTA21:	100177,177777	
003034	101252	125252		DTB21:	101252,125252	
003040	170127	040000		TST21:	LDFPS	#040000 ;CLEAR FLOATING POINT STATUS
003044	172667	177760			LDF	DTA21, AC2 ;LOAD 100177,177777 INTO AC2
003050	170127	047417			LDFPS	#047417 ;LOAD FLOATING POINT STATUS
003054	172267	177754		FPI21:	ADDF	DTB21, AC2 ;ADD 101252,125252 TO AC2
003060	170200				STFPS	FPS ;STORE FLOATING POINT STATUS
003062	022700	047410			CMP	#047410,FPS ;CHECK FLOATING POINT STATUS
003066	001401				BEG	.+4 ;BRANCH IF OK
003070	104000				HLT	;FPS NOT EQUAL TO 047410
003072	174267	175704		STF	AC2, ANS1	;STORE SUM IN ANS1, ANS2
003076	022767	101252	175676	CMP	#101252,ANS1	;CHECK ANS1
003084	001401			BEG	.+4	;BRANCH IF OK
003086	104002			HLT+2		;ANS1 NOT EQUAL TO 101252

```

003110 022767 125252 175666      CMP      #125252,ANS2      :CHECK ANS2
003116 001401      BEQ      .+4              :BRANCH IF OK
003120 104002      HLT+2                    :ANS2 NOT EQUAL TO 125252

```

```

*****
:TEST 22:      TEST ADDF (ADD FLOATING)
:              100177,177777 + 100125,052525 = 100177,177777
:              FPS = 147417,   FSRC = M6-R7,   AC = AC1
:              FEC = 14,       FEA = FPI22
*****

```

```

003122 104400      SCOPE
003124 000404      BR      TS*22           :BRANCH OVER INPUT DATA

```

```

003126 100177 177777      DTA22: 100177,177777
003132 100125 052525      DTB22: 100125,052525

003136 170127 040000      *ST22: LDFPS      #040000      :CLEAR FLOATING POINT STATUS
003142 172567 177750      LDF      DTA22, AC1        :LOAD 100177,177777 INTO AC1
003146 170127 047417      LDFPS      #047417        :LOAD FLOATING POINT STATUS
003152 172167 177754      FPI22:  ADDF      DTB22, AC1 :ADD 100125,052525 TO AC1
003156 170200      STFPS      FPS            :STORE FLOATING POINT STATUS
003160 170367 175636      STST      FEC            :STORE EXCEPTION CODES
003164 022700 147417      CMP      #147417,FPS      :CHECK FLOATING POINT STATUS
003170 001401      BEQ      .+4              :BRANCH IF OK
003172 104000      HLT                    :FPS NOT EQUAL TO 147417

```

```

003174 022767 000014 175660      CMP      #14,   FEC        :CHECK FLOATING EXCEPTION CODE
003202 001401      BEQ      .+4              :BRANCH IF OK
003204 104000      HLT                    :FEC NOT EQUAL TO 14

```

```

003206 022767 003152 175610      CMP      #FPI22, FEA      :CHECK FLOATING EXCEPTION ADDRESS
003214 001401      BEQ      .+4              :BRANCH IF OK
003216 104000      HLT                    :FEA NOT EQUAL TO FPI22

```

```

003220 174167 175556      STF      AC1, ANS1        :STORE SUM IN ANS1, ANS2
003224 022767 100177 175550      CMP      #100177,ANS1     :CHECK ANS1
003232 001401      BEQ      .+4              :BRANCH IF OK
003234 104002      HLT+2                    :ANS1 NOT EQUAL TO 100177

```

```

003236 022767 177777 175540      CMP      #177777,ANS2     :CHECK ANS2
003244 001401      BEQ      .+4              :BRANCH IF OK
003246 104002      HLT+2                    :ANS2 NOT EQUAL TO 177777

```

```

*****
:TEST 23:      TEST ADDF (ADD FLOATING)
:              177652,125252 + 177452,125253 = 100000,000000
:              FPS = 046416,   FSRC = M6-R7,   AC = AC1
*****

```

```

003250 104400      SCOPE
003252 000404      BR      TS*23           :BRANCH OVER INPUT DATA

```

003254	177652	125252		DTA23:	177652,125252	
003260	177452	125253		DTB23:	177452,125253	
003264	170127	046417		TST23:	LDFPS #046417	:LOAD FLOATING POINT STATUS
003270	172567	177760			LDF DTA23, AC1	:LOAD 177652,125252 INTO AC1
003274	172167	177760		FFI23:	ADDF DTB23, AC1	:ADD 177452,125253 TO AC1
003300	170200				STFPS FPS	:STORE FLOATING POINT STATUS
003302	022700	046416			CMP #046416,FPS	:CHECK FLOATING POINT STATUS
003306	001401				BEQ .+4	:BRANCH IF OK
003310	104000				HLT	:FPS NOT EQUAL TO 046416
003312	174167	175464		STF	AC1, ANS1	:STORE SUM IN ANS1, ANS2
003316	022767	100000	175456	CMP	#100000,ANS1	:CHECK ANS1
003324	001401			BEQ	.+4	:BRANCH IF OK
003326	104002			HLT+2		:ANS1 NOT EQUAL TO 100000
003330	022767	000000	175446	CMP	#000000,ANS2	:CHECK ANS2
003336	001401			BEQ	.+4	:BRANCH IF OK
003340	104002			HLT+2		:ANS2 NOT EQUAL TO 000000

:TEST 24: TEST ADDF (ADD FLOATING)
: 000252,125253 + 100425,052525 = 000000,000000
: FPS = 045404, FSRC = M6-R7, AC = ACC

003342	104400			SCOPE		
003344	000404			BR	TST24	:BRANCH OVER INPUT DATA
003346	000252	125253		DTA24:	000252,125253	
003352	100425	052525		DTB24:	100425,052525	
003356	170127	045417		TST24:	LDFPS #045417	:LOAD FLOATING POINT STATUS
003362	172467	177760			LDF DTA24, ACC	:LOAD 000252,125253 INTO ACC
003366	172067	177760		FFI24:	ADDF DTB24, ACC	:ADD 100425,052525 TO ACC
003372	170200				STFPS FPS	:STORE FLOATING POINT STATUS
003374	022700	045404			CMP #045404,FPS	:CHECK FLOATING POINT STATUS
003400	001401				BEQ .+4	:BRANCH IF OK
003402	104000			HLT		:FPS NOT EQUAL TO 045404
003404	174067	175372		STF	ACC, ANS1	:STORE SUM IN ANS1, ANS2
003410	022767	000000	175364	CMP	#000000,ANS1	:CHECK ANS1
003416	001401			BEQ	.+4	:BRANCH IF OK
003420	104002			HLT+2		:ANS1 NOT EQUAL TO 000000
003422	022767	000000	175354	CMP	#000000,ANS2	:CHECK ANS2
003430	001401			BEQ	.+4	:BRANCH IF OK
003432	104002			HLT+2		:ANS2 NOT EQUAL TO 000000

:TEST 25: TEST ADDF (ADD FLOATING)
: 040052,125252 + 040125,052525 = 040300,000000
: FPS = 047400, FSRC = M0-AC3, AC = ACC

```

003434 104400          SCOPE
003436 000404          BR      TST25          :BRANCH OVER INPUT DATA

003440 040052 125252    DTA25: 040052,125252
003444 040125 052525    DTB25: 040125,052525

003450 170127 047417    TST25: LDFPS #047417          :LOAD FLOATING POINT STATUS
003454 172467 177760    LDF      DTA25, AC0          :LOAD 040052,125252 INTO AC0
003460 172767 177760    LDF      DTB25, AC3          :LOAD 040125,052525 INTO AC3
003464 172003          ADDF     AC3, AC0            :ADD AC3 TO AC0
003466 170200          STFPS   FPS                :STORE FLOATING POINT STATUS
003470 022700 047400    CMP      #047400,FPS         :CHECK FLOATING POINT STATUS
003474 001401          BEQ     .+4                 :BRANCH IF OK
003476 104000          HLT     :FPS NOT EQUAL TO 047400

003500 174067 175276          STF      AC0, ANS1          :STORE SUM IN ANS1, ANS2
003504 022767 040300 175270    CMP      #040300,ANS1        :CHECK ANS1
003512 001401          BEQ     .+4                 :BRANCH IF OK
003514 104002          HLT+2   :ANS1 NOT EQUAL TO 040300

003516 022767 000000 175260    CMP      #000000,ANS2        :CHECK ANS2
003524 001401          BEQ     .+4                 :BRANCH IF OK
003526 104002          HLT+2   :ANS2 NOT EQUAL TO 000000

003530 174367 175252          STF      AC3, ANS3          :STORE AC3 IN ANS3, ANS4
003534 022767 040125 175244    CMP      #040125,ANS3        :CHECK ANS3
003542 001401          BEQ     .+4                 :BRANCH IF OK
003544 104004          HLT+4   :AC3 CHANGED

003546 022767 052525 175234    CMP      #052525,ANS4        :CHECK ANS4
003554 001401          BEQ     .+4                 :BRANCH IF OK
003556 104004          HLT+4   :AC3 CHANGED

```

```

*****
:TEST 26: TEST ADDF (ADD FLOATING)
:          000200,000001 + 100200,000000 = 072400,000000
:          FPS = 147400, FSRC = MC-AC0, AC = AC2
:          FEC = 12, FEA = FPI26
*****

```

```

003560 104400          SCOPE
003562 000404          BR      TST26          :BRANCH OVER INPUT DATA

003564 000200 000001    DTA26: 000200,000001
003570 100200 000000    DTB26: 100200,000000

003574 170127 047417    TST26: LDFPS #047417          :LOAD FLOATING POINT STATUS
003600 172667 177760    LDF      DTA26, AC2          :LOAD 000200,000001 INTO AC2
003604 172467 177760    LDF      DTB26, AC0          :LOAD 100200,000000 INTO AC0
003610 172200          ADDF     AC0, AC2            :ADD AC0 TO AC2
003612 170200          STFPS   FPS                :STORE FLOATING POINT STATUS
003614 170367 175202    STST    FEC                :STORE EXCEPTION CODES
003620 022700 147400    CMP      #147400,FPS         :CHECK FLOATING POINT STATUS
003624 001401          BEQ     .+4                 :BRANCH IF OK

```

```

003626 104000          HLT          ;FPS NOT EQUAL TO 147400
003630 022767 000012 175164      CMP      #12,   FEC      ;CHECK FLOATING EXCEPTION CODE
003636 001401          BEQ      .+4          ;BRANCH IF OK
003640 104000          HLT          ;FEC NOT EQUAL TO 12
003642 174267 175134          STF      AC2,   ANS1     ;STORE SUM IN ANS1, ANS2
003646 022767 072400 175126      CMP      #072400,ANS1   ;CHECK ANS1
003654 001401          BEQ      .+4          ;BRANCH IF OK
003656 104002          HLT+2        ;ANS1 NOT EQUAL TO 072400
003660 022767 000000 175116      CMP      #000000,ANS2   ;CHECK ANS2
003666 001401          BEQ      .+4          ;BRANCH IF OK
003670 104002          HLT+2        ;ANS2 NOT EQUAL TO 000000
003672 174067 175110          STF      ACC,   ANS3     ;STORE ACC IN ANS3, ANS4
003676 022767 100200 175102      CMP      #100200,ANS3   ;CHECK ANS3
003704 001401          BEQ      .+4          ;BRANCH IF OK
003706 104004          HLT+4        ;ACC CHANGED
003710 022767 000000 175072      CMP      #000000,ANS4   ;CHECK ANS4
003716 001401          BEQ      .+4          ;BRANCH IF OK
003720 104004          HLT+4        ;ACC CHANGED

```

```

:*****
:TEST 27:      TEST ADD (ADD DOUBLE PRECISION)
:              040000,000000,000000,000000 + 040000,000000,000000,000000 =
:              040200,000000,000000,000000
:              FPS = 047600,   FSRC = M6-R7,   AC = AC2
:*****

```

```

003722 104400          SCOPE
003724 003410          BR      TST27          ;BRANCH OVER INPUT DATA
003726 040000 000000 000000 000000 DTA27: 040000,000000,000000,000000
003734 000000 000000 000000 000000 DTB27: 040000,000000,000000,000000
003736 040000 000000 000000 000000
003744 000000
003746 170127 047617          TST27: LDFPS  #047617      ;LOAD FLOATING POINT STATUS
003752 172667 177750          LDD     DTA27, AC2     ;LOAD 040000,000000,000000,000000 INTO AC2
003756 172267 177754          ADDD   DTB27, AC2     ;ADD 040000,000000,000000,000000 TO AC2
003762 170200          STFPS  FPS           ;STORE FLOATING POINT STATUS
003764 022700 047600          CMP      #047600,FPS   ;CHECK FLOATING POINT STATUS
003770 001401          BEQ      .+4          ;BRANCH IF OK
003772 104000          HLT          ;FPS NOT EQUAL TO 047600
003774 174267 175002          STD     AC2,   ANS1     ;STORE SUM IN ANS1 THRU ANS4
004000 022767 040200 174774      CMP      #040200,ANS1   ;CHECK ANS1
004006 001401          BEQ      .+4          ;BRANCH IF OK
004010 104004          HLT+4        ;ANS1 NOT EQUAL TO 040200
004012 022767 000000 174764      CMP      #000000,ANS2   ;CHECK ANS2
004020 001401          BEQ      .+4          ;BRANCH IF OK
004022 104004          HLT+4        ;ANS2 NOT EQUAL TO 000000

```



```

004024 022767 000000 174754      CMP      #000000,ANS3      ;CHECK ANS3
004032 001401      BEQ      .+4              ;BRANCH IF OK
004034 104004      HLT+4          ;ANS3 NOT EQUAL TO 000000

004036 022767 000000 174744      CMP      #000000,ANS4      ;CHECK ANS4
004044 001401      BEQ      .+4              ;BRANCH IF OK
004046 104004      HLT+4          ;ANS4 NOT EQUAL TO 000000

```

```

:*****
:TEST 30:      TEST ADDD (ADD DOUBLE PRECISION)
:              040177,177777,177777,177777 + 140000,000000,000000,000000 =
:              037777,177777,177777,177776
:              FPS = 047600,   FSRC = M6-R7,   AC = AC1
:*****

```

```

004050 104400      SCOPE
004052 000410      BR      TST30          ;BRANCH OVER INPUT DATA

004054 040177 177777 177777 DTB30: 040177,177777,177777,177777
004062 177777      DTB30: 140000,000000,000000,000000
004064 140000 000000 000000 DTB30: 140000,000000,000000,000000
004072 000000

004074 170127 047617 TST30: LDFPS      #047617      ;LOAD FLOATING POINT STATUS
004100 172567 177750      LDD      DTB30, AC1      ;LOAD 040177,177777,177777,177777 INTO AC1
004104 172167 177754      FPI30:  ADDD      DTB30, AC1      ;ADD 140000,000000,000000,000000 TO AC1
004110 170200      STFPS      FPS          ;STORE FLOATING POINT STATUS
004112 022700 047600      CMP      #047600,FPS      ;CHECK FLOATING POINT STATUS
004116 001401      BEQ      .+4              ;BRANCH IF OK
004120 104000      HLT          ;FPS NOT EQUAL TO 047600

004122 174167 174654      STC      AC1, ANS1        ;STORE SUM IN ANS1 THRU ANS4
004126 022767 037777 174646      CMP      #037777,ANS1      ;CHECK ANS1
004134 001401      BEQ      .+4              ;BRANCH IF OK
004136 104004      HLT+4          ;ANS1 NOT EQUAL TO 037777

004140 022767 177777 174636      CMP      #177777,ANS2      ;CHECK ANS2
004146 001401      BEQ      .+4              ;BRANCH IF OK
004150 104004      HLT+4          ;ANS2 NOT EQUAL TO 177777

004152 022767 177777 174626      CMP      #177777,ANS3      ;CHECK ANS3
004160 001401      BEQ      .+4              ;BRANCH IF OK
004162 104004      HLT+4          ;ANS3 NOT EQUAL TO 177777

004164 022767 177776 174616      CMP      #177776,ANS4      ;CHECK ANS4
004172 001401      BEQ      .+4              ;BRANCH IF OK
004174 104004      HLT+4          ;ANS4 NOT EQUAL TO 177776

```

```

:*****
:TEST 31:      TEST ADDD (ADD DOUBLE PRECISION)
:              040125,052525,052525,052525 + 040125,052525,052525,052525 =
:              040325,052525,052525,052525
:              FPS = 047600,   FSRC = M6-R7,   AC = AC3
:*****

```

M02

MAINDEC-11-DCFPD-C
DCFPD.P11

TEST OF ADDF, ADDD, SUBF, SUBD MACY11 27(732) 17-SEP-76 09:41 PAGE 25
TEST SECTION

```

004176 104400          SCOPE
004200 000410          BR      TST31          ;BRANCH OVER INPUT DATA

004202 040125 052525 052525 DTA31: 040125,052525,052525,052525
004210 052525
004212 040125 052525 052525 DTB31: 040125,052525,052525,052525
004220 052525

004222 170127 047617      TST31: LDFPS  #047617          ;LOAD FLOATING POINT STATUS
004226 172767 177750      LDD      DTA31, AC3          ;LOAD 040125,052525,052525,052525 INTO AC3
004232 172367 177754      FPI31: ADDD   DTB31, AC3          ;ADD 040125,052525,052525,052525 TO AC3
004236 170200      STFPS  FPS                    ;STORE FLOATING POINT STATUS
004240 022700 047600      CMP     #047600,FPS          ;CHECK FLOATING POINT STATUS
004244 001401      BEQ    .+4                    ;BRANCH IF OK
004246 104000      HLT                                ;FPS NOT EQUAL TO 047600

004250 174367 174526      STD     AC3, ANS1            ;STORE SUM IN ANS1 THRU ANS4
004254 022767 040325 174520      CMP     #040325,ANS1        ;CHECK ANS1
004262 001401      BEQ    .+4                    ;BRANCH IF OK
004264 104004      HLT+4                          ;ANS1 NOT EQUAL TO 040325

004266 022767 052525 174510      CMP     #052525,ANS2        ;CHECK ANS2
004274 001401      BEQ    .+4                    ;BRANCH IF OK
004276 104004      HLT+4                          ;ANS2 NOT EQUAL TO 052525

004300 022767 052525 174500      CMP     #052525,ANS3        ;CHECK ANS3
004306 001401      BEQ    .+4                    ;BRANCH IF OK
004310 104004      HLT+4                          ;ANS3 NOT EQUAL TO 052525

004312 022767 052525 17447C      CMP     #052525,ANS4        ;CHECK ANS4
004320 001401      BEQ    .+4                    ;BRANCH IF OK
004322 104004      HLT+4                          ;ANS4 NOT EQUAL TO 052525

```

```

*****
;TEST 32: TEST ADD (ADD DOUBLE PRECISION)
;          040052,125252,125252,125252 + 040125,052525,052525,052524 =
;          040277,177777,177777,177777
;          FPS = 047600, FSRC = M6-R7, AC = AC2
*****

```

```

004324 104400          SCOPE
004326 000410          BR      TST32          ;BRANCH OVER INPUT DATA

004330 040052 125252 125252 DTA32: 040052,125252,125252,125252
004336 125252
004340 040125 052525 052525 DTB32: 040125,052525,052525,052524
004346 052524

004350 170127 047617      TST32: LDFPS  #047617          ;LOAD FLOATING POINT STATUS
004354 172667 177750      LDD      DTA32, AC2          ;LOAD 040052,125252,125252,125252 INTO AC2
004360 172267 177754      FPI32: ADDD   DTB32, AC2          ;ADD 040125,052525,052525,052524 TO AC2
004364 170200      STFPS  FPS                    ;STORE FLOATING POINT STATUS
004366 022700 047600      CMP     #047600,FPS          ;CHECK FLOATING POINT STATUS

```

```

004372 001401      BEQ      .+4      ;BRANCH IF OK
004374 104000      HLT                      ;FPS NOT EQUAL TO 047600

004376 174267 174400  STD      AC2      ANS1      ;STORE SUM IN ANS1 THRU ANS4
004402 022767 040277 174372  CMP      #040277,ANS1      ;CHECK ANS1
004410 001401      BEQ      .+4      ;BRANCH IF OK
004412 104004      HLT+4          ;ANS1 NOT EQUAL TO 040277

004414 022767 177777 174362  CMP      #177777,ANS2      ;CHECK ANS2
004422 001401      BEQ      .+4      ;BRANCH IF OK
004424 104004      HLT+4          ;ANS2 NOT EQUAL TO 177777

004426 022767 177777 174352  CMP      #177777,ANS3      ;CHECK ANS3
004434 001401      BEQ      .+4      ;BRANCH IF OK
004436 104004      HLT+4          ;ANS3 NOT EQUAL TO 177777

004440 022767 177777 174342  CMP      #177777,ANS4      ;CHECK ANS4
004446 001401      BEQ      .+4      ;BRANCH IF OK
004450 104004      HLT+4          ;ANS4 NOT EQUAL TO 177777

```

```

*****
:TEST 33:      TEST ADDD (ADD DOUBLE PRECISION)
:              000200,000000,000000,000000 + 100200,000000,000000,000000 =
:              000000,000000,000000,000000
:              FPS = 047604,   FSRC = M6-R7,   AC = AC2
*****

```

```

004452 :04400      SCOPE
004454 000410      BR      TST33      ;BRANCH OVER INPUT DATA

004456 000200 000000 000000 000000  DTB33: 000200,000000,000000,000000
004464 000000 000000 000000 000000
004466 100200 000000 000000 000000  DTB33: 100200,000000,000000,000000
004474 000000

004476 170127 047617  TST33:  LDFPS  #047617      ;LOAD FLOATING POINT STATUS
004502 172667 177750  LDU      DTB33, AC2      ;LOAD 000200,000000,000000,000000 INTO AC2
004506 172267 177754  FPI33:  ADDD   DTB33, AC2      ;ADD 100200,000000,000000,000000 TO AC2
004512 170200  STFPS   FPS              ;STORE FLOATING POINT STATUS
004514 022700 047604  CMP      #047604,FPS      ;CHECK FLOATING POINT STATUS
004520 001401  BEQ      .+4              ;BRANCH IF OK
004522 104000  HLT                      ;FPS NOT EQUAL TO 047604

004524 174267 174252  STD      AC2      ANS1      ;STORE SUM IN ANS1 THRU ANS4
004530 022767 000000 174244  CMP      #000000,ANS1      ;CHECK ANS1
004536 001401  BEQ      .+4              ;BRANCH IF OK
004540 104004  HLT+4          ;ANS1 NOT EQUAL TO 000000

004542 022767 000000 174234  CMP      #000000,ANS2      ;CHECK ANS2
004550 001401  BEQ      .+4              ;BRANCH IF OK
004552 104004  HLT+4          ;ANS2 NOT EQUAL TO 000000

004554 022767 000000 17422  CMP      #000000,ANS3      ;CHECK ANS3
004562 001401  BEQ      .+4              ;BRANCH IF OK
004564 104004  HLT+4          ;ANS3 NOT EQUAL TO 000000

```

```

004566 022767 000000 174214      CMP      #000000,ANS4      :CHECK ANS4
004574 001401      BEQ      .+4              :BRANCH IF OK
004576 104004      HLT+4                    :ANS4 NOT EQUAL TO 000000

```

```

*****
:TEST 34:      TEST ADD (ADD DOUBLE PRECISION)
:              137525,052525,052525,052525 + 040252,125252,125252,125252 =
:              040217,177777,177777,177777
:              FPS = 047600,   FSRC = M6-R7,   AC = ACC
*****

```

```

004600 104400      SCOPE
004602 000410      BR      TST34           :BRANCH OVER INPUT DATA

```

```

004604 137525 052525 052525 0TA34: 137525,052525,052525,052525
004612 052525
004614 040252 125252 125252 0TB34: 040252,125252,125252,125252
004622 125252

```

```

004624 170127 047617  TST34:  LDFFS  #047617      :LOAD FLOATING POINT STATUS
004630 172467 177750  LD      0TA34, ACC  :LOAD 137525,052525,052525,052525 INTO ACC
004634 172067 177754  FPI34:  ACCD   0TB34, ACC  :ADD 040252,125252,125252,125252 TO ACC
004640 170200  STFFS  FPS         :STORE FLOATING POINT STATUS
004642 022700 047600  CMP      #047600,FPS  :CHECK FLOATING POINT STATUS
004646 001401      BEQ      .+4          :BRANCH IF OK
004650 104000      HLT+4                    :FPS NOT EQUAL TO 047600

```

```

004652 174067 174124  STC      ACC, ANS1   :STORE SUM IN ANS1 THRU ANS4
004656 022767 040217 174116  CMP      #040217,ANS1 :CHECK ANS1
004664 001401      BEQ      .+4          :BRANCH IF OK
004666 104004      HLT+4                    :ANS1 NOT EQUAL TO 040217

```

```

004670 022767 177777 174106  CMP      #177777,ANS2 :CHECK ANS2
004676 001401      BEQ      .+4          :BRANCH IF OK
004700 104004      HLT+4                    :ANS2 NOT EQUAL TO 177777

```

```

004702 022767 177777 174076  CMP      #177777,ANS3 :CHECK ANS3
004710 001401      BEQ      .+4          :BRANCH IF OK
004712 104004      HLT+4                    :ANS3 NOT EQUAL TO 177777

```

```

004714 022767 177777 174066  CMP      #177777,ANS4 :CHECK ANS4
004722 001401      BEQ      .+4          :BRANCH IF OK
004724 104004      HLT+4                    :ANS4 NOT EQUAL TO 177777

```

```

*****
:TEST 35:      TEST ADD (ADD DOUBLE PERCISION)
:              140252,125252,125252,125252 + 037653 =
:              140177,152525,052525,052524
:              FPS = 047610,   FSRC = M2-R7,   AC = ACC
*****

```

```

004726 104400      SCOPE
004728 000410      BR      TST35           :BRANCH OVER INPUT DATA

```

```

004732 140252 :25252 125252 DTB35: 140252,125252,125252,125252
004740 125252

004742 170127 047617 TST35: LDFPS #047617 :LOAD FLOATING POINT STATUS
004746 172667 177750 LDD DTB35 AC2 :LOAD 140252,125252,125252,125252 INTO AC2
004752 172227 037653 FPI35: ADDD #.33733,AC2 :ADD 037653 TO AC2
004756 170200 STFPS FPS :STORE FLOATING POINT STATUS
004760 022700 047610 CMP #047610,FPS :CHECK FLOATING POINT STATUS
004764 001401 BEQ .+4 :BRANCH IF OK
004768 104000 HLT :FPS NOT EQUAL TO 047610

004770 174267 174006 STD AC2 ANS1 :STORE SUM IN ANS1 THRU ANS4
004774 022767 140177 174000 CMP #140177,ANS1 :CHECK ANS1
005002 001401 BEQ .+4 :BRANCH IF OK
005004 104004 HLT+4 :ANS1 NOT EQUAL TO 140177

005006 022767 :52525 173770 CMP #152525,ANS2 :CHECK ANS2
005014 001401 BEQ .+4 :BRANCH IF OK
005016 104004 HLT+4 :ANS2 NOT EQUAL TO 152525

005020 022767 052525 173760 CMP #052525,ANS3 :CHECK ANS3
005026 001401 BEQ .+4 :BRANCH IF OK
005030 104004 HLT+4 :ANS3 NOT EQUAL TO 052525

005032 022767 052524 173750 CMP #052524,ANS4 :CHECK ANS4
005040 001401 BEQ .+4 :BRANCH IF OK
005042 104004 HLT+4 :ANS4 NOT EQUAL TO 052524

```

```

*****
TEST 36: TEST ADDD (ADD DOUBLE PRECISION)
100200,000000,000000,000000 + 077777,177777,177777,177777 =
077777,177777,177777,177777
FPS = 047600, FSRC = M6-R7, AC = AC2
*****

```

```

005044 104400 SCOPE
005046 000410 BR TST36 :BRANCH OVER INPUT DATA

005050 100200 000000 000000 DTB36: 100200,000000,000000,000000
005056 000000
005060 077777 177777 177777 DTB36: 077777,177777,177777,177777
005066 177777

005070 170127 047617 TST35: LDFPS #047617 :LOAD FLOATING POINT STATUS
005074 172667 177750 LDD DTB36 AC2 :LOAD 100200,000000,000000,000000 INTO AC2
005100 172267 177754 FPI36: ADDD DTB36 AC2 :ADD 077777,177777,177777,177777 TO AC2
005104 170200 STFPS FPS :STORE FLOATING POINT STATUS
005106 022700 047600 CMP #047600,FPS :CHECK FLOATING POINT STATUS
005112 001401 BEQ .+4 :BRANCH IF OK
005114 104000 HLT :FPS NOT EQUAL TO 047600

005116 174267 173660 STD AC2 ANS1 :STORE SUM IN ANS1 THRU ANS4
005120 022767 077777 173652 CMP #077777,ANS1 :CHECK ANS1
005124 001401 BEQ .+4 :BRANCH IF OK

```

005132	104004			HLT+4		:ANS1 NOT EQUAL TO 077777
005134	022767	177777	173642	CMP	#177777,ANS2	:CHECK ANS2
005142	001401			BEQ	.+4	:BRANCH IF OK
005144	104004			HLT+4		:ANS2 NOT EQUAL TO 177777
005146	022767	177777	173632	CMP	#177777,ANS3	:CHECK ANS3
005154	001401			BEQ	.+4	:BRANCH IF OK
005156	104004			HLT+4		:ANS3 NOT EQUAL TO 177777
005160	022767	177777	173622	CMP	#177777,ANS4	:CHECK ANS4
005166	001401			BEQ	.+4	:BRANCH IF OK
005170	104004			HLT+4		:ANS4 NOT EQUAL TO 177777

```

*****
TEST 37:      TEST ADDD (ADD DOUBLE PRECISION)
              177777,177777,177777,177777 + 100200,000000,000000,000000 =
              177777,177777,177777,177777
              FPS = 047610,  FSRC = M6-R7,  AC = AC2
*****

```

005172	104400			SCOPE		
005174	000410			BR	TST37	:BRANCH OVER INPUT DATA
005176	177777	177777	177777	DTA37:	177777,177777,177777,177777	
005204	177777					
005206	100200	000000	000000	DTB37:	100200,000000,000000,000000	
005214	000000					
005216	170127	047617		TST37:	LDFPS	#047617
005222	172667	177750		LDD	DTA37, AC2	:LOAD 177777,177777,177777,177777 INTO AC2
005226	172267	177754		FP137:	ADD	DTB37, AC2
005232	170200			S*FPS	FPS	:STORE FLOATING POINT STATUS
005234	022700	047610		CMP	#047610,FPS	:CHECK FLOATING POINT STATUS
005240	001401			BEQ	.+4	:BRANCH IF OK
005242	104000			HLT		:FPS NOT EQUAL TO 047610
005244	174267	173532		STD	AC2, ANS1	:STORE SUM IN ANS1 THRU ANS4
005250	022767	177777	173524	CMP	#177777,ANS1	:CHECK ANS1
005256	001401			BEQ	.+4	:BRANCH IF OK
005260	104004			HLT+4		:ANS1 NOT EQUAL TO 177777
005262	022767	177777	173514	CMP	#177777,ANS2	:CHECK ANS2
005270	001401			BEQ	.+4	:BRANCH IF OK
005272	104004			HLT+4		:ANS2 NOT EQUAL TO 177777
005274	022767	177777	173504	CMP	#177777,ANS3	:CHECK ANS3
005302	001401			BEQ	.+4	:BRANCH IF OK
005304	104004			HLT+4		:ANS3 NOT EQUAL TO 177777
005306	022767	177777	173474	CMP	#177777,ANS4	:CHECK ANS4
005314	001401			BEQ	.+4	:BRANCH IF OK
005316	104004			HLT+4		:ANS4 NOT EQUAL TO 177777

E03

MAINDEC-11-DCFPD-C
DCFPD.P11

TEST OF ADDF, ADDD, SUBF, SUBC MACY11 27(732) 17-SEP-76 09:41 PAGE 30
TEST SECTION

:TEST 40: TEST ADD (ADD DOUBLE PRECISION)
: 147125.052525,052525,052525 + 031152,125252,125252,125252 =
: 147125.052525,052525,052524
: FPS = 047610, FSRC = M6-R7, AC = AC3

005320	104400			SCOPE			
005320	000410			BR	TST40		:BRANCH OVER INPUT DATA
005324	147125	052525	052525	DTA40:	147125,052525,052525,052525		
005324	052525						
005324	031152	125252	125252	DTB40:	031152,125252,125252,125252		
005342	125252						
005344	170127	047617		TST40:	LDFPS	#047617	:LOAD FLOATING POINT STATUS
005350	172767	177750		LD	DTA40, AC3		:LOAD 147125,052525,052525,052525,052525,052525,052525,052525
005354	172367	177754		ADD	DTB40, AC3		:ADD 031152,125252,125252,125252,052525,052525,052525,052525
005360	170200			STFPS	FPS		:STORE FLOATING POINT STATUS
005362	022700	047610		CMP	#047610,FPS		:CHECK FLOATING POINT STATUS
005366	001401			BEG	++		:BRANCH IF OK
005370	104000			HLT			:FPS NOT EQUAL TO 047610
005372	174367	173404		STO	ANS1		:STORE SUM IN ANS1 THRU ANS4
005376	022767	147125	173376	CMP	#147125,ANS1		:CHECK ANS1
005404	001401			BEG	++		:BRANCH IF OK
005406	104004			HLT	++		:ANS1 NOT EQUAL TO 147125
005410	022767	052525	173366	CMP	#052525,ANS2		:CHECK ANS2
005416	001401			BEG	++		:BRANCH IF OK
005420	104004			HLT	++		:ANS2 NOT EQUAL TO 052525
005422	022767	052525	173356	CMP	#052525,ANS3		:CHECK ANS3
005428	001401			BEG	++		:BRANCH IF OK
005432	104004			HLT	++		:ANS3 NOT EQUAL TO 052525
005434	022767	052524	173346	CMP	#052524,ANS4		:CHECK ANS4
005440	001401			BEG	++		:BRANCH IF OK
005444	104004			HLT	++		:ANS4 NOT EQUAL TO 052524

:TEST 41: TEST ADD (ADD DOUBLE PRECISION)
: 031152,125252,125252,125252 + 146725,052525,052525,052525 =
: 146725,052525,052525,052523
: FPS = 047610, FSRC = M6-R7, AC = AC3

005446	104400			SCOPE			
005446	000410			BR	TST41		:BRANCH OVER INPUT DATA
005450	031152	125252	125252	DTA41:	031152,125252,125252,125252		
005450	052525	052525	052525	DTB41:	146725,052525,052525,052525		

F03

MANDEC-11-00FPD-C
00FPD.P11

TEST OF ADDF, ADD, SJBF, SUBD
TEST SECTION

MACY11 27(732) 17-SEP-76 09:41 PAGE 31

```

005502 170127 047617 TST41: LDFPS #047617 :LOAD FLOATING POINT STATUS
005504 172567 177750 LDD DTA41, AC3 :LOAD 031152,125252,125252,125252 INTO AC3
005506 172367 177754 FPI41: ADDD DTB41, AC3 :ADD 146725,052525,052525,052525 TO AC3
005510 170200 STFPS FPS :STORE FLOATING POINT STATUS
005514 022700 047610 CMP #047610,FPS :CHECK FLOATING POINT STATUS
005516 001401 BEQ .+4 :BRANCH IF OK
005516 104000 HLT :FPS NOT EQUAL TO 047610

005520 174367 173256 STD AC3, ANS1 :STORE SUM IN ANS1 THRU ANS4
005524 022767 146725 173250 CMP #146725,ANS1 :CHECK ANS1
005532 001401 BEQ .+4 :BRANCH IF OK
005534 104004 HLT+4 :ANS1 NOT EQUAL TO 146725

005536 022767 052525 173240 CMP #052525,ANS2 :CHECK ANS2
005544 001401 BEQ .+4 :BRANCH IF OK
005546 104004 HLT+4 :ANS2 NOT EQUAL TO 052525

005550 022767 052525 173230 CMP #052525,ANS3 :CHECK ANS3
005556 001401 BEQ .+4 :BRANCH IF OK
005560 104004 HLT+4 :ANS3 NOT EQUAL TO 052525

005562 022767 052523 173220 CMP #052523,ANS4 :CHECK ANS4
005570 001401 BEQ .+4 :BRANCH IF OK
005572 104004 HLT+4 :ANS4 NOT EQUAL TO 052523

```

```

*****
:TEST 42: TEST ADD (ADD DOUBLE PRECISION)
: 147125,052525,052525,052525 + 131036 =
: 147125,052525,052525,052526
: FPS = 047610, FSRC = M2-R7, AC = AC1
*****

```

```

005574 104400 SCOPE
005576 000404 BR TST42 :BRANCH OVER INPUT DATA

005600 147125 052525 052525 DTA42: 147125,052525,052525,052525
005606 052525

005610 170127 047617 TST42: LDFPS #047617 :LOAD FLOATING POINT STATUS
005614 172567 177760 LDD DTA42, AC1 :LOAD 147125,052525,052525,052525 INTO AC1
005620 172127 131036 FPI42: ADDD # -23E-10,AC1 :ADD 131036 TO AC1
005624 170200 STFPS FPS :STORE FLOATING POINT STATUS
005626 022700 047610 CMP #047610,FPS :CHECK FLOATING POINT STATUS
005632 001401 BEQ .+4 :BRANCH IF OK
005634 104000 HLT :FPS NOT EQUAL TO 047610

005636 174167 173140 STD AC1, ANS1 :STORE SUM IN ANS1 THRU ANS4
005642 022767 147125 173132 CMP #147125,ANS1 :CHECK ANS1
005650 001401 BEQ .+4 :BRANCH IF OK
005652 104004 HLT+4 :ANS1 NOT EQUAL TO 147125

005654 022767 052525 173122 CMP #052525,ANS2 :CHECK ANS2
005662 001401 BEQ .+4 :BRANCH IF OK
005664 104004 HLT+4 :ANS2 NOT EQUAL TO 052525

```



```

005666 022767 052525 173112      CMP      #052525,ANS3      ;CHECK ANS3
005674 001401      BEQ      .+4              ;BRANCH IF OK
005676 104004      HLT+4          ;ANS3 NOT EQUAL TO 052525

005700 022767 052526 173102      CMP      #052526,ANS4      ;CHECK ANS4
005706 001401      BEQ      .+4              ;BRANCH IF OK
005710 104004      HLT+4          ;ANS4 NOT EQUAL TO 052526

```

```

*****
TEST 43:      TEST ADD (ADD DOUBLE PRECISION)
              147125.052525,052525,052525 + 030752,125252,125252,125252 =
              147125.052525,052525,052525
FPS = 047610, FSRC = M6-R7, AC = ACC
*****

```

```

005712 104400      SCOPE
005714 000410      BR      TS*43          ;BRANCH OVER INPUT DATA

005716 147125 052525 052525 052525 DTA43: 147125,052525,052525,052525
005724 052525
005726 030752 125252 125252 DTB43: 030752,125252,125252,125252
005734 125252

```

```

005736 170127 047617      TS*43: LDFPS      #047617      ;LOAD FLOATING POINT STATUS
005742 172467 177750      LDD      DTA43, ACC      ;LOAD 147125,052525,052525,052525 INTO ACC
005746 172067 177754      ADD      DTB43, ACC      ;ADD 030752,125252,125252,125252 TO ACC
005752 170200      STFPS      FPS          ;STORE FLOATING POINT STATUS
005754 022700 047610      CMP      #047610,FPS      ;CHECK FLOATING POINT STATUS
005760 001401      BEQ      .+4              ;BRANCH IF OK
005762 104000      HLT          ;FPS NOT EQUAL TO 047610

```

```

005764 174067 173012      STD      ACC, ANS1       ;STORE SUM IN ANS1 THRU ANS4
005770 022767 147125 173004      CMP      #147125,ANS1     ;CHECK ANS1
005776 001401      BEQ      .+4              ;BRANCH IF OK
006000 104004      HLT+4          ;ANS1 NOT EQUAL TO 147125

```

```

006002 022767 052525 172774      CMP      #052525,ANS2     ;CHECK ANS2
006010 001401      BEQ      .+4              ;BRANCH IF OK
006012 104004      HLT+4          ;ANS2 NOT EQUAL TO 052525

```

```

006014 022767 052525 172764      CMP      #052525,ANS3     ;CHECK ANS3
006022 001401      BEQ      .+4              ;BRANCH IF OK
006024 104004      HLT+4          ;ANS3 NOT EQUAL TO 052525

```

```

006026 022767 052525 172754      CMP      #052525,ANS4     ;CHECK ANS4
006034 001401      BEQ      .+4              ;BRANCH IF OK
006036 104004      HLT+4          ;ANS4 NOT EQUAL TO 052525

```

```

*****
TEST 44:      TEST ADD (ADD DOUBLE PRECISION)
              077452,125252,125252,125253 + 077652,125252,125252,125252 =
              000000,000000,000000,000000
FPS = 147606, FSRC = M6-R7, AC = ACC
FEI = 10, FEI = FEI+4
*****

```

```

006040 104400          SCOPE
006042 000410          BR      TST44      :BRANCH OVER INPUT DATA

006044 077452 125252 125252 DTA44: 077452,125252,125252,125253
006052 125253
006054 077652 125252 125252 DTB44: 077652,125252,125252,125252
006062 125252

006064 170127 047617 TST44: LDFPS #047617 :LOAD FLOATING POINT STATUS
006070 172667 177750          LDD  DTA44, AC2 :LOAD 077452,125252,125252,125253 INTO AC2
006074 172267 177754          ADD  DTB44, AC2 :ADD 077652,125252,125252,125252 TO AC2
006100 170200          STFPS FPS :STORE FLOATING POINT STATUS
006102 170367 172714          STST  FEC :STORE EXCEPTION CODES
006106 022700 147606          CMP  #147606,FPS :CHECK FLOATING POINT STATUS
006112 001401          BEQ  .+4 :BRANCH IF OK
006114 104000          HLT  :FPS NOT EQUAL TO 147606

006116 022767 000010 172676          CMP  #10, FEC :CHECK FLOATING EXCEPTION CODE
006124 001401          BEQ  .+4 :BRANCH IF OK
006126 104000          HLT  :FEC NOT EQUAL TO 10

006130 022767 006074 172666          CMP  #FPI44, FEA :CHECK FLOATING EXCEPTION ADDRESS
006136 001401          BEQ  .+4 :BRANCH IF OK
006140 104000          HLT  :FEA NOT EQUAL TO FPI44

006142 174267 172634          STD  AC2, ANS1 :STORE SUM IN ANS1 THRU ANS4
006146 022767 000000 172626          CMP  #000000,ANS1 :CHECK ANS1
006154 001401          BEQ  .+4 :BRANCH IF OK
006156 104004          HLT  +4 :ANS1 NOT EQUAL TO 000000

006160 022767 000000 172616          CMP  #000000,ANS2 :CHECK ANS2
006166 001401          BEQ  .+4 :BRANCH IF OK
006170 104004          HLT  +4 :ANS2 NOT EQUAL TO 000000

006172 022767 000000 172606          CMP  #000000,ANS3 :CHECK ANS3
006200 001401          BEQ  .+4 :BRANCH IF OK
006202 104004          HLT  +4 :ANS3 NOT EQUAL TO 000000

006204 022767 000000 172576          CMP  #000000,ANS4 :CHECK ANS4
006212 001401          BEQ  .+4 :BRANCH IF OK
006214 104004          HLT  +4 :ANS4 NOT EQUAL TO 000000

```

```

*****
TEST 45: TEST ADDD (ADD DOUBLE PRECISION)
          177652,125252,125252,125252 + 177452,125252,125252,125253 =
          100000,000000,000000,000000
          FPS = 147616, FSRC = M6-R7, AC = AC2
          FEC = 10, FEA = FPI45
*****

```

```

006216 104400          SCOPE
006218 000410          BR      TST45      :BRANCH OVER INPUT DATA

```

```

006222 177652 125252 125252 DT45: 177652,125252,125252,125252
006230 125252
006232 177452 125252 125252 DT845: 177452,125252,125252,125253
006240 125253

006242 170127 047617 TST45: LDFPS #047617 :LOAD FLOATING POINT STATUS
006246 172667 177750 LDC DT45, AC2 :LOAD 177652,125252,125252,125252 INTO AC2
006252 172267 177754 FPI45: ADD DT845, AC2 :ADD 177452,125252,125252,125253 TO AC2
006256 170200 STFPS FPS :STORE FLOATING POINT STATUS
006260 170367 172536 STST FEC :STORE EXCEPTION CODES
006264 022700 147616 CMP #147616,FPS :CHECK FLOATING POINT STATUS
006270 001401 BEQ .+4 :BRANCH IF OK
006272 104000 HLT :FPS NOT EQUAL TO 147616

006274 022767 000010 172520 CMP #10, FEC :CHECK FLOATING EXCEPTION CODE
006302 001401 BEQ .+4 :BRANCH IF OK
006304 104000 HLT :FEC NOT EQUAL TO 10

006306 022767 006252 172510 CMP #FPI45, FEA :CHECK FLOATING EXCEPTION ADDRESS
006314 001401 BEQ .+4 :BRANCH IF OK
006316 104000 HLT :FEA NOT EQUAL TO FPI45

006320 174267 172456 172450 STD AC2,ANS1 :STORE SUM IN ANS1 THRU ANS4
006324 022767 100000 172450 CMP #100000,ANS1 :CHECK ANS1
006332 001401 BEQ .+4 :BRANCH IF OK
006334 104004 HLT+4 :ANS1 NOT EQUAL TO 100000

006336 022767 000000 172440 CMP #000000,ANS2 :CHECK ANS2
006344 001401 BEQ .+4 :BRANCH IF OK
006346 104004 HLT+4 :ANS2 NOT EQUAL TO 000000

006350 022767 000000 172430 CMP #000000,ANS3 :CHECK ANS3
006356 001401 BEQ .+4 :BRANCH IF OK
006360 104004 HLT+4 :ANS3 NOT EQUAL TO 000000

006362 022767 000000 172420 CMP #000000,ANS4 :CHECK ANS4
006370 001401 BEQ .+4 :BRANCH IF OK
006372 104004 HLT+4 :ANS4 NOT EQUAL TO 000000

```

```

*****
:TEST 46: TEST ADD (ADD DOUBLE PRECISION)
: 000425,052525,052525,052525 + 100252,125252,125252,125253 =
: 000177,177777,177777,177776
: FPS = 147604, FSRC = M6-R7, AC = AC1
: FEC = 12, FEA = FPI46
*****

```

```

006374 104000 SCOPE
006376 000410 BR TST46 :BRANCH OVER INPUT DATA

006400 000425 052525 052525 DT46: 000425,052525,052525,052525
006406 052525
006412 125252 125252 DT846: 100252,125252,125252,125253
006418 125253

```

```

006420 170127 047617 TST46: LDFPS #047617 ;LOAD FLOATING POINT STATUS
006424 172567 177750 LDD DTA46, AC1 ;LOAD 000425,052525,052525,052525 INTO AC1
006430 172167 177754 FPI46: ADDD DTB46, AC1 ;ADD 100252,125252,125252,125253 TO AC1
006434 170200 STFPS FPS ;STORE FLOATING POINT STATUS
006436 170367 172360 STST FEC ;STORE EXCEPTION CODES
006442 022700 147604 CMP #147604,FPS ;CHECK FLOATING POINT STATUS
006446 001401 BEQ .+4 ;BRANCH IF OK
006450 104000 HLT ;FPS NOT EQUAL TO 147604

006452 022767 000012 172342 CMP #12, FEC ;CHECK FLOATING EXCEPTION CODE
006460 001401 BEQ .+4 ;BRANCH IF OK
006462 104000 HLT ;FEC NOT EQUAL TO 12

006464 022767 006430 172332 CMP #FPI46, FEA ;CHECK FLOATING EXCEPTION ADDRESS
006472 001401 BEQ .+4 ;BRANCH IF OK
006474 104000 HLT ;FEA NOT EQUAL TO FPI46

006476 174167 172300 STD AC1, ANS1 ;STORE SUM IN ANS1 THRU ANS4
006502 022767 000177 172272 CMP #000177,ANS1 ;CHECK ANS1
006510 001401 BEQ .+4 ;BRANCH IF OK
006512 104004 HLT+4 ;ANS1 NOT EQUAL TO 000177

006514 022767 177777 172262 CMP #177777,ANS2 ;CHECK ANS2
006522 001401 BEQ .+4 ;BRANCH IF OK
006524 104004 HLT+4 ;ANS2 NOT EQUAL TO 177777

006526 022767 177777 172252 CMP #177777,ANS3 ;CHECK ANS3
006534 001401 BEQ .+4 ;BRANCH IF OK
006536 104004 HLT+4 ;ANS3 NOT EQUAL TO 177777

006540 022767 177776 172242 CMP #177776,ANS4 ;CHECK ANS4
006546 001401 BEQ .+4 ;BRANCH IF OK
006550 104004 HLT+4 ;ANS4 NOT EQUAL TO 177776

```

```

:*****
:TEST 47: TEST ADD (ADD DOUBLE PRECISION)
: 000252,125252,125252,125253 + 100425,052525,052525,052525 =
: 100177,177777,177777,177776
: FPS = 147614, FSRC = M6-R7, AC = AC3
: FEC = 12, FEA = FPI47
:*****

```

```

006552 104400 SCOPE
006554 000410 BR TST47 ;BRANCH OVER INPUT DATA

006556 000252 125252 125252 DTA47: 000252,125252,125252,125253
006564 125253 DTB47: 100425,052525,052525,052525
006566 100425 052525 052525
006574 052525

006576 170127 047617 TST47: LDFPS #047617 ;LOAD FLOATING POINT STATUS
006602 172767 177750 LDD DTA47, AC3 ;LOAD 000252,125252,125252,125253 INTO AC3
006606 172367 177754 FPI47: ADDD DTB47, AC3 ;ADD 100425,052525,052525,052525 TO AC3
006612 170200 STFPS FPS ;STORE FLOATING POINT STATUS
006614 170367 172252 STST FEC ;STORE EXCEPTION CODES

```

K03

MAINDEC-11-DCFPD-C
DCFPD.F11

TEST OF ADDF, ADDD, SJBF, SUBD
TEST SECTION

MACY11 27(732) 17-SEP-76 09:41 PAGE 36

```

006620 022700 147614      CMP      #147614,FPS      :CHECK FLOATING POINT STATUS
006624 001401      BEQ      .+4          :BRANCH IF OK
006626 104000      HLT                      :FPS NOT EQUAL TO 147614

006630 022767 000012 172164      CMP      #12,      FEC      :CHECK FLOATING EXCEPTION CODE
006636 001401      BEQ      .+4          :BRANCH IF OK
006640 104000      HLT                      :FEC NOT EQUAL TO 12

006642 022767 006606 172154      CMP      #FPI47, FEA      :CHECK FLOATING EXCEPTION ADDRESS
006650 001401      BEQ      .+4          :BRANCH IF OK
006652 104000      HLT                      :FEA NOT EQUAL TO FPI47

006654 174367 172122      STD      AC3,      ANS1      :STORE SUM IN ANS1 THRU ANS4
006660 022767 100177 172114      CMP      #100177,ANS1      :CHECK ANS1
006666 001401      BEQ      .+4          :BRANCH IF OK
006670 104004      HLT+4                :ANS1 NOT EQUAL TO 100177

006672 022767 177777 172104      CMP      #177777,ANS2      :CHECK ANS2
006700 001401      BEQ      .+4          :BRANCH IF OK
006702 104004      HLT+4                :ANS2 NOT EQUAL TO 177777

006704 022767 177777 172074      CMP      #177777,ANS3      :CHECK ANS3
006712 001401      BEQ      .+4          :BRANCH IF OK
006714 104004      HLT+4                :ANS3 NOT EQUAL TO 177777

006716 022767 177776 172064      CMP      #177776,ANS4      :CHECK ANS4
006724 001401      BEQ      .+4          :BRANCH IF OK
006726 104004      HLT+4                :ANS4 NOT EQUAL TO 177776

```

```

*****
:TEST 50:      TEST ADDD (ADD DOUBLE PRECISION)
:              101252,125252,125252,125252 + 100177,177777,177777,177777 =
:              101252,125252,125252,125252
:              FPS = 147610,      FSRC = M6-R7,      AC = AC2
:              FEC = 14,      FEA = FPI50
*****

```

```

006730 104400      SCOPE
006732 000410      BR      TST50      ;BRANCH OVER INPUT DATA

006734 101252 125252 125252 DTA50: 101252,125252,125252,125252
006742 125252
006744 100177 177777 177777 DTB50: 100177,177777,177777,177777
006752 177777

006754 170127 047617      TST50: LDFPS      #047617      :LOAD FLOATING POINT STATUS
006760 172667 177750      LDD      DTA50, AC2      :LOAD 101252,125252,125252,125252 INTO AC2
006764 172267 177754      FPIS0: ADDD      DTB50, AC2      :ADD 100177,177777,177777,177777 TO AC2
006770 170200      STFPS      FPS          :STORE FLOATING POINT STATUS
006772 170367 172024      STST      FEC          :STORE EXCEPTION CODES
006776 022700 147610      CMP      #147610,FPS      :CHECK FLOATING POINT STATUS
007002 001401      BEQ      .+4          :BRANCH IF OK
007004 104000      HLT                      :FPS NOT EQUAL TO 147610

007006 022767 000014 172006      CMP      #14,      FEC      :CHECK FLOATING EXCEPTION CODE

```

```

007014 001401      BEQ      .+4      ;BRANCH IF OK
007016 104000      HLT
007020 022767 006764 171776  CMP      #FPI50, FEA ;CHECK FLOATING EXCEPTION ADDRESS
007026 001401      BEQ      .+4      ;BRANCH IF OK
007030 104000      HLT      ;FEA NOT EQUAL TO FPI50

007032 174267 171744      STD      AC2, ANS1 ;STORE SUM IN ANS1 THRU ANS4
007036 022767 101252 171736  CMP      #101252, ANS1 ;CHECK ANS1
007044 001401      BEQ      .+4      ;BRANCH IF OK
007046 104004      HLT+4     ;ANS1 NOT EQUAL TO 101252

007050 022767 125252 171726  CMP      #125252, ANS2 ;CHECK ANS2
007056 001401      BEQ      .+4      ;BRANCH IF OK
007060 104004      HLT+4     ;ANS2 NOT EQUAL TO 125252

007062 022767 125252 171716  CMP      #125252, ANS3 ;CHECK ANS3
007070 001401      BEQ      .+4      ;BRANCH IF OK
007072 104004      HLT+4     ;ANS3 NOT EQUAL TO 125252

007074 022767 125252 171706  CMP      #125252, ANS4 ;CHECK ANS4
007102 001401      BEQ      .+4      ;BRANCH IF OK
007104 104004      HLT+4     ;ANS4 NOT EQUAL TO 125252

```

```

*****
*EST 51: TEST ADDO (ADD DOUBLE PRECISION)
*          077452, 125252, 125252, 125253 + 077652, 125252, 125252, 125252 =
*          000000, 000000, 000000, 000000
*          FPS = 046606, FSRC = M6-R7, AC = AC3
*****

```

```

007106 104400      SCOPE
007110 000410      BR      TST51      ;BRANCH OVER INPUT DATA

007112 077452 125252 125252 DTAS1: 077452, 125252, 125252, 125253
007120 125253
007122 077652 125252 125252 DTB51: 077652, 125252, 125252, 125252
007130 125252

007132 170127 046617      TST51: LDFPS #046617 ;LOAD FLOATING POINT STATUS
007136 172767 177750      LDD      DTAS1, AC3 ;LOAD 077452, 125252, 125252, 125253 INTO AC3
007142 172367 177754      FPIS1: ADD      DTB51, AC3 ;ADD 077652, 125252, 125252, 125252 TO AC3
007146 170200      STFPS  FPS ;STORE FLOATING POINT STATUS
007150 022700 046606      CMP      #046606, FPS ;CHECK FLOATING POINT STATUS
007154 001401      BEQ      .+4      ;BRANCH IF OK
007156 104000      HLT      ;FPS NOT EQUAL TO 046606

007160 174367 171616      STD      AC3, ANS1 ;STORE SUM IN ANS1 THRU ANS4
007164 022767 000000 171610  CMP      #000000, ANS1 ;CHECK ANS1
007172 001401      BEQ      .+4      ;BRANCH IF OK
007174 104004      HLT+4     ;ANS1 NOT EQUAL TO 000000

007176 022767 000000 171600  CMP      #000000, ANS2 ;CHECK ANS2
007204 001401      BEQ      .+4      ;BRANCH IF OK
007206 104004      HLT+4     ;ANS2 NOT EQUAL TO 000000

```

M03

MAINDEC-11-DCFPD-C
DCFPD.F11

TEST OF ADDF, ADDD, SJB, SUBD MACY11 27(732) 17-SEP-76 09:41 PAGE 38
TEST SECTION

```

007210 022767 000000 171570      CMP      #000000,ANS3      ;CHECK ANS3
007216 001401      BEQ      .+4              ;BRANCH IF OK
007220 104004      HLT+4          ;ANS3 NOT EQUAL TO 000000

007222 022767 000000 171560      CMP      #000000,ANS4      ;CHECK ANS4
007230 001401      BEQ      .+4              ;BRANCH IF OK
007232 104004      HLT+4          ;ANS4 NOT EQUAL TO 000000

```

```

:*****
:TEST 52:      TEST ADDD (ADD DOUBLE PRECISION)
:              000425,052525,052525,052525 + 100252,125252,125252,125253 =
:              000000,000000,000000,000000
:              FPS = 045604,   FSRC = M6-R7,   AC = FC3
:*****

```

```

007234 104400      SCOPE
007236 000410      BR      TST52          ;BRANCH OVER INPUT DATA

007240 000425 052525 052525 DTAS2: 000425,052525,052525,052525
007246 052525
007250 100252 125252 125252 DTBS2: 100252,125252,125252,125253
007256 125253

007260 170127 045617      TST52: LDFPS      #045617      ;LOAD FLOATING POINT STATUS
007264 172767 177750      LDD      DTAS2, AC3      ;LOAD 000425,052525,052525,052525 INTO AC3
007270 172367 177754      FPIS2: ADDD      DTBS2, AC3      ;ADD 100252,125252,125252,125253 TO AC3
007274 170200      STFPS      FPS          ;STORE FLOATING POINT STATUS
007276 022700 045604      CMP      #045604,FPS      ;CHECK FLOATING POINT STATUS
007302 001401      BEQ      .+4              ;BRANCH IF OK
007304 104000      HLT          ;FPS NOT EQUAL TO 045604

007306 174367 171470      STD      AC3, ANS1        ;STORE SUM IN ANS1 THRU ANS4
007312 022767 000000 171462      CMP      #000000,ANS1      ;CHECK ANS1
007320 001401      BEQ      .+4              ;BRANCH IF OK
007322 104004      HLT+4          ;ANS1 NOT EQUAL TO 000000

007324 022767 000000 171452      CMP      #000000,ANS2      ;CHECK ANS2
007332 001401      BEQ      .+4              ;BRANCH IF OK
007334 104004      HLT+4          ;ANS2 NOT EQUAL TO 000000

007336 022767 000000 171442      CMP      #000000,ANS3      ;CHECK ANS3
007344 001401      BEQ      .+4              ;BRANCH IF OK
007346 104004      HLT+4          ;ANS3 NOT EQUAL TO 000000

007350 022767 000000 171432      CMP      #000000,ANS4      ;CHECK ANS4
007356 001401      BEQ      .+4              ;BRANCH IF OK
007360 104004      HLT+4          ;ANS4 NOT EQUAL TO 000000

```

```

:*****
:TEST 53:      TEST ADDD (ADD DOUBLE PRECISION)
:              000000,000000,000000,000000 + 000000,000000,000000,000000 =
:              000000,000000,000000,000000
:              FPS = 047604,   FSRC = M0-AC3,   AC = AC2
:*****

```

```

007362 104400          SCOPE
007364 000410          BR      TST53          ;BRANCH OVER INPUT DATA

007366 000000 000000 000000 DTA53: 000000,000000,000000,000000
007374 000000
007376 000000 000000 000000 DTB53: 000000,000000,000000,000000
007404 000000

007406 170127 047617      TST53: LDFPS  #047617          ;LOAD FLOATING POINT STATUS
007412 172667 177750      LDD  DTA53, AC2          ;LOAD 000000,000000,000000,000000 INTO AC2
007416 172767 177754      LDD  DTB53, AC3          ;LOAD 000000,000000,000000,000000 INTO AC3
007422 172203          ADDD  AC3, AC2          ;ADD AC3 TO AC2
007424 170200          STFPS FPS              ;STORE FLOATING POINT STATUS
007426 022700 047604      CMP  #047604,FPS        ;CHECK FLOATING POINT STATUS
007432 001401          BEQ  .+4              ;BRANCH IF OK
007434 104000          HLT

007436 174267 171340          STD  AC2, ANS1          ;STORE SUM IN ANS1 THRU ANS4
007442 022767 000000 171332      CMP  #000000,ANS1      ;CHECK ANS1
007450 001401          BEQ  .+4              ;BRANCH IF OK
007452 104004          HLT+4              ;ANS1 NOT EQUAL TO 000000

007454 022767 000000 171322      CMP  #000000,ANS2      ;CHECK ANS2
007462 001401          BEQ  .+4              ;BRANCH IF OK
007464 104004          HLT+4              ;ANS2 NOT EQUAL TO 000000

007466 022767 000000 171312      CMP  #000000,ANS3      ;CHECK ANS3
007474 001401          BEQ  .+4              ;BRANCH IF OK
007476 104004          HLT+4              ;ANS3 NOT EQUAL TO 000000

007500 022767 000000 171302      CMP  #000000,ANS4      ;CHECK ANS4
007506 001401          BEQ  .+4              ;BRANCH IF OK
007510 104004          HLT+4              ;ANS4 NOT EQUAL TO 000000

007512 174367 171274          STD  AC3, ANS5          ;STORE AC3 IN ANS5 THRU ANS8
007516 022767 000000 171266      CMP  #000000,ANS5      ;CHECK ANS5
007524 001401          BEQ  .+4              ;BRANCH IF OK
007526 104010          HLT+10             ;AC3 CHANGED

007530 022767 000000 171256      CMP  #000000,ANS6      ;CHECK ANS6
007536 001401          BEQ  .+4              ;BRANCH IF OK
007540 104010          HLT+10             ;AC3 CHANGED

007542 022767 000000 171246      CMP  #000000,ANS7      ;CHECK ANS7
007550 001401          BEQ  .+4              ;BRANCH IF OK
007552 104010          HLT+10             ;AC3 CHANGED

007554 022767 000000 171236      CMP  #000000,ANS8      ;CHECK ANS8
007562 001401          BEQ  .+4              ;BRANCH IF OK
007564 104010          HLT+10             ;AC3 CHANGED

```

TEST 54: TEST ADDD ADD DOUBLE PRECISION)

: 077777,177777,177777,177777 + 077777,177777,177777,177777 =
: 000177,177777,177777,177777
: FPS = 147606, FSAC = MO-AC2, AC = AC1
: FEC = 10, FEA = FP154
:*****

007566	104400			SCOPE			
007570	000410			BR	TST54		: BRANCH OVER INPUT DATA
007572	077777	177777	177777	DTA54:	077777,177777,177777,177777		
007600	177777						
007602	077777	177777	177777	DTB54:	077777,177777,177777,177777		
007610	177777						
007612	170127	047617		TST54:	LDFPS	#047617	: LOAD FLOATING POINT STATUS
007616	172567	177750			LDD	DTA54, AC1	: LOAD 077777,177777,177777,177777 INTO AC1
007622	172667	177754			LDD	DTB54, AC2	: LOAD 077777,177777,177777,177777 INTO AC2
007626	172102				ADD	AC2, AC1	: ADD AC2 TO AC1
007630	170200				S,FPS	FPS	: STORE FLOATING POINT STATUS
007632	170367	171164			STST	FEC	: STORE EXCEPTION CODES
007636	022700	147606			CMP	#147606,FPS	: CHECK FLOATING POINT STATUS
007642	001401				BEG	.+4	: BRANCH IF OK
007644	104000				HLT		: FPS NOT EQUAL TO 147606
007646	022767	000010	171146		CMP	#10, FEC	: CHECK FLOATING EXCEPTION CODE
007654	001401				BEG	.+4	: BRANCH IF OK
007656	104000				HLT		: FEC NOT EQUAL TO 10
007660	174167	171116			STD	AC1, ANS1	: STORE SLM IN ANS1 THRU ANS4
007664	022767	000177	171110		CMP	#000177,ANS1	: CHECK ANS1
007672	001401				BEG	.+4	: BRANCH IF OK
007674	104004				HLT+4		: ANS1 NOT EQUAL TO 000177
007676	022767	177777	171100		CMP	#177777,ANS2	: CHECK ANS2
007704	001401				BEG	.+4	: BRANCH IF OK
007706	104004				HLT+4		: ANS2 NOT EQUAL TO 177777
007710	022767	177777	171070		CMP	#177777,ANS3	: CHECK ANS3
007716	001401				BEG	.+4	: BRANCH IF OK
007720	104004				HLT+4		: ANS3 NOT EQUAL TO 177777
007722	022767	177777	171060		CMP	#177777,ANS4	: CHECK ANS4
007730	001401				BEG	.+4	: BRANCH IF OK
007732	104004				HLT+4		: ANS4 NOT EQUAL TO 177777
007734	174267	171052			STD	AC2, ANS5	: STORE AC2 IN ANS5 THRU ANS6
007740	022767	077777	171044		CMP	#077777,ANS5	: CHECK ANS5
007746	001401				BEG	.+4	: BRANCH IF OK
007750	104010				HLT+10		: AC2 CHANGED
007752	022767	177777	171034		CMP	#177777,ANS6	: CHECK ANS6
007760	001401				BEG	.+4	: BRANCH IF OK
007762	104010				HLT+10		: AC2 CHANGED
007764	022767	177777	171024		CMP	#177777,ANS7	: CHECK ANS7
007772	001401				BEG	.+4	: BRANCH IF OK

```

007774 104010          HLT+10          :AC2 CHANGED
007776 022767 177777 171014  CMP      #177777,ANS8  :CHECK ANS8
010004 001401          BEQ      .+4          :BRANCH IF OK
010006 104010          HLT+10          :AC2 CHANGED

```

```

*****
TEST S5:      TEST SUBF (SUBTRACT FLOATING)
              040177,177777 - 040000,000000 = 037777,177776
              FPS = 047400,   FSRC = M6-R7,   AC = AC1
*****

```

```

010010 104400          SCOPE
010012 000404          BR      TST55          :BRANCH OVER INPUT DATA

```

```

010014 040177 177777  DTAS5: 040177,177777
010020 040000 000000  DTB55: 040000,000000

TST55:  LDFPS  #047417          :LOAD FLOATING POINT STATUS
        LDF   DTAS5, AC1       :LOAD 040177,177777 INTO AC1
        SUBF  DTB55, AC1       :SUBTRACT 040000,000000 FROM AC1
        STFPS FPS              :STORE FLOATING POINT STATUS
        CMP   #047400,FPS      :CHECK FLOATING POINT STATUS
        BEQ   .+4              :BRANCH IF OK
        HLT   +2              :FPS NOT EQUAL TO 047400

```

```

010052 174167 170724  STS   AC1,ANS1          :STORE DIFFERENCE IN ANS1, ANS2
010056 022767 037777 170716  CMP   #037777,ANS1      :CHECK ANS1
010064 001401          BEQ   .+4              :BRANCH IF OK
010066 104002          HLT   +2              :ANS1 NOT EQUAL TO 037777

```

```

010070 022767 177776 170705  CMP   #177776,ANS2      :CHECK ANS2
010076 001401          BEQ   .+4              :BRANCH IF OK
010100 104002          HLT   +2              :ANS2 NOT EQUAL TO 177776

```

```

*****
TEST S6:      TEST SUBF (SUBTRACT FLOATING)
              040125,052525 - 140125,052525 = 040325,052525
              FPS = 047400,   FSRC = M6-R7,   AC = AC1
*****

```

```

010102 104400          SCOPE
010104 000404          BR      TST56          :BRANCH OVER INPUT DATA

```

```

010106 040125 052525  DTAS6: 040125,052525
010112 140125 052525  DTB56: 140125,052525

TST56:  LDFPS  #047417          :LOAD FLOATING POINT STATUS
        LDF   DTAS6, AC1       :LOAD 040125,052525 INTO AC1
        SUBF  DTB56, AC1       :SUBTRACT 140125,052525 FROM AC1
        STFPS FPS              :STORE FLOATING POINT STATUS
        CMP   #047400,FPS      :CHECK FLOATING POINT STATUS
        BEQ   .+4              :BRANCH IF OK
        HLT   +2              :FPS NOT EQUAL TO 047400

```

MANDEE-1-20FPC-0
207P21.011

TEST OF ADDF, ADDO, SUBF, SUBO MACY11 27(732) 17-SEP-76 09:41 PAGE 42
*TEST SECTION

010144	174167	170632		STF	AC1	ANS1	:STORE DIFFERENCE IN ANS1, ANS2
010150	022767	040325	170624	CMP	#040325,ANS1		:CHECK ANS1
010156	001401			BEQ	.+4		:BRANCH IF OK
010160	104002			HLT+2			:ANS1 NOT EQUAL TO 040325
010162	022767	052525	170614	CMP	#052525,ANS2		:CHECK ANS2
010168	001401			BEQ	.+4		:BRANCH IF OK
010172	104002			HLT+2			:ANS2 NOT EQUAL TO 052525

```

*****
:TEST 57: TEST SUBF (SUBTRACT FLOATING)
:          040052,125252 - 140125,052524 = 040277,177777
:          FPS = 047400, FSRC = M6-R7, AC = AC3
*****

```

010174	104400			SCOPE			
010176	000404			BR	TST57		:BRANCH OVER INPUT DATA
010200	040052	125252		DTA57:	040052,125252		
010204	140125	052524		DTB57:	140125,052524		
010210	170127	047417		TST57:	LDFPS	#047417	:LOAD FLOATING POINT STATUS
010214	172767	177760		LDF	DTA57, AC3		:LOAD 040052,125252 INTO AC3
010220	173367	177760		FP157:	SUBF	DTB57, AC3	:SUBTRACT 140125,052524 FROM AC3
010224	170200			STFPS	FPS		:STORE FLOATING POINT STATUS
010226	022700	047400		CMP	#047400,FPS		:CHECK FLOATING POINT STATUS
010232	001401			BEQ	.+4		:BRANCH IF OK
010234	104000			HLT			:FPS NOT EQUAL TO 047400
010236	174367	170540		STF	AC3	ANS1	:STORE DIFFERENCE IN ANS1, ANS2
010242	022767	040277	170532	CMP	#040277,ANS1		:CHECK ANS1
010250	001401			BEQ	.+4		:BRANCH IF OK
010252	104002			HLT+2			:ANS1 NOT EQUAL TO 040277
010254	022767	177777	170522	CMP	#177777,ANS2		:CHECK ANS2
010260	001401			BEQ	.+4		:BRANCH IF OK
010264	104002			HLT+2			:ANS2 NOT EQUAL TO 177777

```

*****
:TEST 60: TEST SUBF (SUBTRACT FLOATING)
:          000200,000000 - 000200,000000 = 000000,000000
:          FPS = 047404, FSRC = M6-R7, AC = AC1
*****

```

010266	104400			SCOPE			
010270	000404			BR	TST60		:BRANCH OVER INPUT DATA
010272	000200	000000		DTA60:	000200,000000		
010276	000200	000000		DTB60:	000200,000000		
010302	170127	047417		TST60:	LDFPS	#047417	:LOAD FLOATING POINT STATUS
010306	172767	177760		LDF	DTA60, AC1		:LOAD 000200,000000 INTO AC1
010310	173367	177760		FP160:	SUBF	DTB60, AC1	:SUBTRACT 000200,000000 FROM AC1

MACY11 27.732) 17-SEP-76 09:41 PAGE 43

TEST OF ADDF, ADDD, SUBF, SUBD TEST SECTION

010316	170200			STF	FPS		:STORE FLOATING POINT STATUS
010318	022700	047404		CMP	#047404.FPS		:CHECK FLOATING POINT STATUS
010324	001401			BEQ	+.4		:BRANCH IF OK
010326	104000			HLT			:FPS NOT EQUAL TO 047404
010330	174167	170446		STF	AC1	ANS1	:STORE DIFFERENCE IN ANS1, ANS2
010334	022767	000000	170440	CMP	#000000.ANS1		:CHECK ANS1
010340	001401			BEQ	+.4		:BRANCH IF OK
010344	104002			HLT+2			:ANS1 NOT EQUAL TO 000000
010348	022767	000000	170430	CMP	#000000.ANS2		:CHECK ANS2
010354	001401			BEQ	+.4		:BRANCH IF OK
010356	104002			HLT+2			:ANS2 NOT EQUAL TO 000000

```

*****
:TEST 61: TEST SUBF (SUBTRACT FLOATING)
:          137525.052525 - 140252.125252 = 040217.177777
:          FPS = 047400, FSRC = M6-R7, AC = AC3
*****

```

010360	104400			SCOPE			
010362	000404			BR	TS*61		:BRANCH OVER INPUT DATA
010364	137525	052525		TS*61:	137525.052525		
010370	140252	125252		D*651:	140252.125252		
010374	170127	047417		TS*61:	LOFPS	#047417	:LOAD FLOATING POINT STATUS
010400	172767	177760			LOF	DT*61, AC3	:LOAD 137525.052525 INTO AC3
010404	173367	177760		FP*61:	SUBF	DT*61, AC3	:SUBTRACT 140252.125252 FROM AC3
010410	170200				STFPS	FPS	:STORE FLOATING POINT STATUS
010412	022700	047400			CMP	#047400.FPS	:CHECK FLOATING POINT STATUS
010416	001401				BEQ	+.4	:BRANCH IF OK
010420	104000				HLT		:FPS NOT EQUAL TO 047400
010422	174367	170354		STF	AC3	ANS1	:STORE DIFFERENCE IN ANS1, ANS2
010426	022767	040217	170346	CMP	#040217.ANS1		:CHECK ANS1
010434	001401			BEQ	+.4		:BRANCH IF OK
010436	104002			HLT+2			:ANS1 NOT EQUAL TO 040217
010440	022767	177777	170336	CMP	#177777.ANS2		:CHECK ANS2
010446	001401			BEQ	+.4		:BRANCH IF OK
010450	104002			HLT+2			:ANS2 NOT EQUAL TO 177777

```

*****
:TEST 62: TEST SUBF (SUBTRACT FLOATING)
:          140252.125252 - 137525.052525 = 140217.177777
:          FPS = 047410, FSRC = M6-R7, AC = AC1
*****

```

010454	104400			SCOPE			
010456	000404			BR	TS*62		:BRANCH OVER INPUT DATA
010460	140252	125252		TS*62:	140252.125252		
010462	137525	052525		FP*62:	137525.052525		

010466	170127	047417	TST62:	LDFPS	#047417	:LOAD FLOATING POINT STATUS
010472	172567	177760		LDF	DTA62, AC1	:LOAD 140252,125252 INTO AC1
010476	173167	177760	FP162:	SUBF	DTB62, AC1	:SUBTRACT 137525,052525 FROM AC1
010502	170200			STFPS	FPS	:STORE FLOATING POINT STATUS
010504	022700	047410		CMP	#047410,FPS	:CHECK FLOATING POINT STATUS
010510	001401			BEG	+.4	:BRANCH IF OK
010512	104000			HLT		:FPS NOT EQUAL TO 047410
010514	174167	170262		STF	AC1, ANS1	:STORE DIFFERENCE IN ANS1, ANS2
010520	022767	140217	170254	CMP	#140217,ANS1	:CHECK ANS1
010526	001401			BEG	+.4	:BRANCH IF OK
010530	104002			HLT+2		:ANS1 NOT EQUAL TO 140217
010532	022767	177777	170244	CMP	#177777,ANS2	:CHECK ANS2
010540	001401			BEG	+.4	:BRANCH IF OK
010542	104002			HLT+2		:ANS2 NOT EQUAL TO 177777

```

*****
:TEST 63: TEST SUBF (SUBTRACT FLOATING)
:          100200,000000 - 177777,177777 = 077777,177777
:          FPS = 047400, FSRC = M6-R7, AC = ACC
*****

```

010544	104400			SCOPE		
010546	000404			BR	*ST63	:BRANCH OVER INPUT DATA
010550	100200	000000	DTA63:	100200,000000		
010554	177777	177777	DTB63:	177777,177777		
010560	170127	047417	*ST63:	LDFPS	#047417	:LOAD FLOATING POINT STATUS
010564	172467	177760		LDF	DTA63, ACC	:LOAD 100200,000000 INTO ACC
010570	173067	177760	FP163:	SUBF	DTB63, ACC	:SUBTRACT 177777,177777 FROM ACC
010574	170200			STFPS	FPS	:STORE FLOATING POINT STATUS
010576	022700	047400		CMP	#047400,FPS	:CHECK FLOATING POINT STATUS
010602	001401			BEG	+.4	:BRANCH IF OK
010604	104000			HLT		:FPS NOT EQUAL TO 047400
010606	174067	170170		STF	ACC, ANS1	:STORE DIFFERENCE IN ANS1, ANS2
010612	022767	077777	170162	CMP	#077777,ANS1	:CHECK ANS1
010620	001401			BEG	+.4	:BRANCH IF OK
010622	104002			HLT+2		:ANS1 NOT EQUAL TO 077777
010624	022767	177777	170152	CMP	#177777,ANS2	:CHECK ANS2
010630	001401			BEG	+.4	:BRANCH IF OK
010632	104002			HLT+2		:ANS2 NOT EQUAL TO 177777

```

*****
:TEST 64: TEST SUBF (SUBTRACT FLOATING)
:          177777,177777 - 000200,000000 = 177777,177777
:          FPS = 047410, FSRC = M6-R7, AC = ACC
*****

```

010636	104400			SCOPE		
--------	--------	--	--	-------	--	--

```

010640 000404 BR TST64 :BRANCH OVER INPUT DATA
010642 177777 177777 DTB64: 177777,177777
010646 000200 000000 DTB64: 000200,000000
010652 170127 047417 TST64: LDFPS #047417 :LOAD FLOATING POINT STATUS
010656 172667 177760 LDF DTB64, ACC :LOAD 177777,177777 INTO ACC
010660 173267 177760 FP164: SUBF DTB64, ACC :SUBTRACT 000200,000000 FROM ACC
010664 170200 STFPS FPS :STORE FLOATING POINT STATUS
010668 022700 047410 CMP #047410, FPS :CHECK FLOATING POINT STATUS
010672 001401 BEQ .+4 :BRANCH IF OK
010676 104000 HLT :FPS NOT EQUAL TO 047410

010700 174267 170076 STF ACC, ANS1 :STORE DIFFERENCE IN ANS1, ANS2
010704 022767 177777 170070 CMP #177777, ANS1 :CHECK ANS1
010712 001401 BEQ .+4 :BRANCH IF OK
010714 104002 HLT +2 :ANS1 NOT EQUAL TO 177777

010716 022767 177777 170080 CMP #177777, ANS2 :CHECK ANS2
010720 001401 BEQ .+4 :BRANCH IF OK
010724 104002 HLT +2 :ANS2 NOT EQUAL TO 177777

```

```

*****
:TEST 65: TEST SUBF (SUBTRACT FLOATING)
:         143125,052525 - 035152,125252 = 143125,052526
:         FPS = 047410, FSRC = ME-R7, AC = ACC
*****

```

```

010730 104400 SCOPE
010732 000404 BR TST65 :BRANCH OVER INPUT DATA
010734 143125 052525 DTB65: 143125,052525
010740 035152 125252 DTB65: 035152,125252
010744 170127 047417 TST65: LDFPS #047417 :LOAD FLOATING POINT STATUS
010750 172467 177760 LDF DTB65, ACC :LOAD 143125,052525 INTO ACC
010754 173067 177760 FP165: SUBF DTB65, ACC :SUBTRACT 035152,125252 FROM ACC
010760 170200 STFPS FPS :STORE FLOATING POINT STATUS
010764 022700 047410 CMP #047410, FPS :CHECK FLOATING POINT STATUS
010768 001401 BEQ .+4 :BRANCH IF OK
010770 104000 HLT :FPS NOT EQUAL TO 047410

010772 174067 170004 STF ACC, ANS1 :STORE DIFFERENCE IN ANS1, ANS2
010776 022767 143125 167776 CMP #143125, ANS1 :CHECK ANS1
010804 001401 BEQ .+4 :BRANCH IF OK
010806 104002 HLT +2 :ANS1 NOT EQUAL TO 143125

010810 022767 052526 167766 CMP #052526, ANS2 :CHECK ANS2
010814 001401 BEQ .+4 :BRANCH IF OK
010818 104002 HLT +2 :ANS2 NOT EQUAL TO 052526

```

```

*****
:TEST 66: TEST SUBF (SUBTRACT FLOATING)
:         143125,052525 - 134752,125252 = 143125,052526
*****

```

:***** FPS = 047410, FSRC = M6-R7, AC = AC1 *****

```

011022 104400          SCOPE
011024 000404          BR      TST66          :BRANCH OVER INPUT DATA

011026 143125 052525  DTAB6: 143125.052525
011032 134752 125252  DTB66: 134752.125252

011036 170127 047417  TST66: LDFPS  #047417          :LOAD FLOATING POINT STATUS
011042 172567 177760  LDF  DTAB6, AC1          :LOAD 143125.052525 INTO AC1
011046 173167 177760  FPI66: SUBF  DTB66, AC1          :SUBTRACT 134752.125252 FROM AC1
011052 170200  STFPS  FPS              :STORE FLOATING POINT STATUS
011054 022700 047410  CMP  #047410.FPS          :CHECK FLOATING POINT STATUS
011060 001401  BEQ  .+4                :BRANCH IF OK
011062 104000  HLT

011064 174167 167712  STF  AC1, ANS1          :STORE DIFFERENCE IN ANS1, ANS2
011070 022767 143125 167704  CMP  #143125.ANS1        :CHECK ANS1
011076 001401  BEQ  .+4                :BRANCH IF OK
011100 104002  HLT+2                :ANS1 NOT EQUAL TO 143125

011102 022767 052525 167674  CMP  #052525.ANS2        :CHECK ANS2
011110 001401  BEQ  .+4                :BRANCH IF OK
011112 104002  HLT+2                :ANS2 NOT EQUAL TO 052525

```

:*****
:TEST 67: TEST SUBF (SUBTRACT FLOATING)
: 077452.125253 - 177652.125252 = 000000.000000
: FPS = 147406, FSRC = M6-R7, AC = AC2
: FEC = 10, FEA = FPI67
:*****

```

011114 104400          SCOPE
011116 000404          BR      TST67          :BRANCH OVER INPUT DATA

011120 077452 125253  DTAB7: 077452.125253
011124 177652 125252  DTB67: 177652.125252

011130 170127 047417  TST67: LDFPS  #047417          :LOAD FLOATING POINT STATUS
011134 172667 177760  LDF  DTAB7, AC2          :LOAD 077452.125253 INTO AC2
011140 173267 177760  FPI67: SUBF  DTB67, AC2          :SUBTRACT 177652.125252 FROM AC2
011144 170200  STFPS  FPS              :STORE FLOATING POINT STATUS
011146 170367 167650  STS  FEC              :STORE EXCEPTION CODES
011152 022700 147406  CMP  #147406.FEC          :CHECK FLOATING POINT STATUS
011156 001401  BEQ  .+4                :BRANCH IF OK
011160 104000  HLT+2                :FEC NOT EQUAL TO 147406

011162 022767 000010 167632  CMP  #10, FEC           :CHECK FLOATING EXCEPTION CODE
011170 001401  BEQ  .+4                :BRANCH IF OK
011172 104000  HLT+2                :FEC NOT EQUAL TO 10

011174 022767 011140 167622  CMP  #FPI67, FEA        :CHECK FLOATING EXCEPTION ADDRESS
011180 001401  BEQ  .+4                :BRANCH IF OK
011182 104000  HLT+2                :FEA NOT EQUAL TO FPI67

```

```

011206 174267 167570      STF      ACC, ANS1      :STORE DIFFERENCE IN ANS1, ANS2
011212 022767 000000 167562  CMP      #000000,ANS1  :CHECK ANS1
011220 001401      BEQ      .+4           :BRANCH IF OK
011222 104002      HLT+2       :ANS1 NOT EQUAL TO 000000

011234 022767 000000 167552  CMP      #000000,ANS2  :CHECK ANS2
011236 001401      BEQ      .+4           :BRANCH IF OK
011238 104002      HLT+2       :ANS2 NOT EQUAL TO 000000

```

```

*****
:TEST 70:      TEST SUBF (SUBTRACT FLOATING)
:              177652,125252 - 077452,125253 = 100000,000000
:              FPS = 147416,   FSRC = M6-R7,   AC = ACC
:              FEC = 10,     FEA = FPI70
*****

```

```

011236 104400      SCOPE
011240 000404      BR      TST70      :BRANCH OVER INPUT DATA

011242 177652 125252  DTA70: 177652,125252
011246 077452 125253  DTB70: 077452,125253

011252 170127 047417  TST70: LDFPS  #047417      :LOAD FLOATING POINT STATUS
011256 172467 177760      LDF      DTA70, ACC    :LOAD 177652,125252 INTO ACC
011262 173067 177760      SUBF     DTB70, ACC    :SUBTRACT 077452,125253 FROM ACC
011266 170200      STFPS   FPS          :STORE FLOATING POINT STATUS
011270 170367 167526  STST    FEC          :STORE EXCEPTION CODES
011274 022700 147416  CMP      #147416,FPS   :CHECK FLOATING POINT STATUS
011300 001401      BEQ      .+4           :BRANCH IF OK
011302 104000      HLT          :FPS NOT EQUAL TO 147416

011304 022767 000010 167510  CMP      #10,   FEC    :CHECK FLOATING EXCEPTION CODE
011312 001401      BEQ      .+4           :BRANCH IF OK
011314 104000      HLT          :FEC NOT EQUAL TO 10

011316 022767 011262 167500  CMP      #FPI70, FEA   :CHECK FLOATING EXCEPTION ADDRESS
011324 001401      BEQ      .+4           :BRANCH IF OK
011326 104000      HLT          :FEA NOT EQUAL TO FPI70

011330 174067 167446      STF      ACC, ANS1      :STORE DIFFERENCE IN ANS1, ANS2
011334 022767 100000 167440  CMP      #100000,ANS1  :CHECK ANS1
011342 001401      BEQ      .+4           :BRANCH IF OK
011344 104002      HLT+2       :ANS1 NOT EQUAL TO 100000

011346 022767 000000 167430  CMP      #000000,ANS2  :CHECK ANS2
011354 001401      BEQ      .+4           :BRANCH IF OK
011356 104002      HLT+2       :ANS2 NOT EQUAL TO 000000

```

```

*****
:TEST 71:      TEST SUBF (SUBTRACT FLOATING)
:              000425,052525 - 000252,125253 = 000173,177776
:              FPS = 147404,   FSRC = M6-R7,   AC = ACC
:              FEC = 12,     FEA = FPI71
*****

```

011360	104400			SCOPE			
011362	000404			BR	TST71		:BRANCH OVER INPUT DATA
011364	000425	052525		DTA71:	000425,052525		
011370	000252	125253		DTB71:	000252,125253		
011374	170127	047417		TST71:	LDFPS	#047417	:LOAD FLOATING POINT STATUS
011400	172467	177760			LDF	DTA71, ACC	:LOAD 000425,052525 INTO ACC
011404	173067	177760		FPI71:	SUBF	DTB71, ACC	:SUBTRACT 000252,125253 FROM ACC
011410	170200				STFPS	FPS	:STORE FLOATING POINT STATUS
011412	170367	167404			STST	FEC	:STORE EXCEPTION CODES
011415	022700	147404			CMP	#147404,FPS	:CHECK FLOATING POINT STATUS
011422	001401				BEQ	.+4	:BRANCH IF OK
011424	104000				HLT		:FPS NOT EQUAL TO 147404
011426	022767	000012	167366		CMP	#12, FEC	:CHECK FLOATING EXCEPTION CODE
011434	001401				BEQ	.+4	:BRANCH IF OK
011436	104000				HLT		:FEC NOT EQUAL TO 12
011440	022767	011404	167356		CMP	#FPI71, FEA	:CHECK FLOATING EXCEPTION ADDRESS
011446	001401				BEQ	.+4	:BRANCH IF OK
011450	104000				HLT		:FEA NOT EQUAL TO FPI71
011452	174067	167324			STF	ACC, ANS1	:STORE DIFFERENCE IN ANS1, ANS2
011456	022767	000177	167316		CMP	#000177,ANS1	:CHECK ANS1
011464	001401				BEQ	.+4	:BRANCH IF OK
011466	104002				HLT	+2	:ANS1 NOT EQUAL TO 000177
011470	022767	177776	167306		CMP	#177776,ANS2	:CHECK ANS2
011476	001401				BEQ	.+4	:BRANCH IF OK
011500	104002				HLT	+2	:ANS2 NOT EQUAL TO 177776

```

TEST 72: TEST SUBF (SUBTRACT FLOATING)
          000252,125253 - 000425,052525 = 100177,177776
          FPS = 147414, FSRC = M6-R7, AC = AC1
          FEC = 12, FEA = FPI72
  
```

011502	104400			SCOPE			
011504	000404			BR	TST72		:BRANCH OVER INPUT DATA
011506	000252	125253		DTA72:	000252,125253		
011512	000425	052525		DTB72:	000425,052525		
011516	170127	047417		TST72:	LDFPS	#047417	:LOAD FLOATING POINT STATUS
011522	172567	177760			LDF	DTA72, AC1	:LOAD 000252,125253 INTO AC1
011528	173167	177760		FPI72:	SUBF	DTB72, AC1	:SUBTRACT 000425,052525 FROM AC1
011534	170200				STFPS	FPS	:STORE FLOATING POINT STATUS
011540	170367	167262			STST	FEC	:STORE EXCEPTION CODES
011546	022700	147414			CMP	#147414,FPS	:CHECK FLOATING POINT STATUS
011552	001401				BEQ	.+4	:BRANCH IF OK
011558	104002				HLT		:FPS NOT EQUAL TO 147414

011744 104002

HLT+2

;ANS2 NOT EQUAL TO 125252

:TEST 74: TEST SUBF (SUBTRACT FLOATING)
: 077452,125253 - 177652,125252 = 000000,000000
: FPS = 046406, FSRC = M6-R7, AC = AC1

011746 104400
011750 000404

SCOPE
BR TST74

;BRANCH OVER INPUT DATA

011752 077452 125253
011755 177652 125252

DTA74: 077452,125253
DTB74: 177652,125252

011762 170127 046417
011766 172567 177760
011772 173167 177760
011776 170200
012000 022700 046406
012004 001401
012006 104000

TST74: LDFPS #046417
LDF DTA74, AC1
FPI74: SUBF DTB74, AC1
STFPS FPS
CMP #046406,FPS
BEQ .+4
HLT

;LOAD FLOATING POINT STATUS
;LOAD 077452,125253 INTO AC1
;SUBTRACT 177652,125252 FROM AC1
;STORE FLOATING POINT STATUS
;CHECK FLOATING POINT STATUS
;BRANCH IF OK
;FPS NOT EQUAL TO 046406

012010 174167 166766
012014 022767 000000 166760
012022 001401
012024 104002

STF AC1, ANS1
CMP #000000,ANS1
BEQ .+4
HLT+2

;STORE DIFFERENCE IN ANS1, ANS2
;CHECK ANS1
;BRANCH IF OK
;ANS1 NOT EQUAL TO 000000

012026 022767 000000 166750
012034 001401
012036 104002

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

;CHECK ANS2
;BRANCH IF OK
;ANS2 NOT EQUAL TO 000000

:TEST 75: TEST SUBF (SUBTRACT FLOATING)
: 000425,052525 - 000252,125253 = 000000,000000
: FPS = 045404, FSRC = M6-R7, AC = AC2

012040 104400
012042 000404

SCOPE
BR TST75

;BRANCH OVER INPUT DATA

012044 000425 052525
012050 000252 125253

DTA75: 000425,052525
DTB75: 000252,125253

012054 170127 045417
012060 172667 177760
012064 173267 177760
012070 170200
012072 022700 045404
012076 001401
012100 104000

TST75: LDFPS #045417
LDF DTA75, AC2
FPI75: SUBF DTB75, AC2
STFPS FPS
CMP #045404,FPS
BEQ .+4
HLT

;LOAD FLOATING POINT STATUS
;LOAD 000425,052525 INTO AC2
;SUBTRACT 000252,125253 FROM AC2
;STORE FLOATING POINT STATUS
;CHECK FLOATING POINT STATUS
;BRANCH IF OK
;FPS NOT EQUAL TO 045404

012102 174267 166674
012106 022767 000000 166666
012114 001401

STF AC2, ANS1
CMP #000000,ANS1
BEQ .+4

;STORE DIFFERENCE IN ANS1, ANS2
;CHECK ANS1
;BRANCH IF OK

MO4

MAINDEC-11-DCFPD-3
DCFPD.P11

TEST OF ADDF, ADDD, SJB, SUBD MACY11 27(732) 17-SEP-76 09:41 PAGE 51
TEST SECTION

012116	104002			HLT+2		;ANS1 NOT EQUAL TO 000000
012120	022767	000000	166656	CMP	#000000,ANS2	;CHECK ANS2
012126	001401			BEQ	.+4	;BRANCH IF OK
012130	104002			HLT+2		;ANS2 NOT EQUAL TO 000000

```

:*****
:TEST 76:      TEST SUBF (SUBTRACT FLOATING)
:              000000,000000 - 040000,000000 = 140000,000000
:              FPS = 047410,   FSRC = MO-AC2,   AC = AC3
:*****

```

012132	104400			SCOPE		
012134	000404			BR	TST76	;BRANCH OVER INPUT DATA
012136	000000	000000		DTA76:	000000,000000	
012142	040000	000000		DTB76:	040000,000000	
012146	170127	047417		TST76:	LDFPS	#047417
012152	172767	177760			LDF	CTA76, AC3
012156	172667	177760			LDF	DTB76, AC2
012162	173302				SLBF	AC2, AC3
012164	170200				STFPS	FPS
012166	022700	047410			CMP	#047410,FPS
012172	001401				BEQ	.+4
012174	104000				HLT	;FPS NOT EQUAL TO 047410
012176	174367	166600		STF	AC3, ANS1	;STORE DIFFERENCE IN ANS1, ANS2
012202	022767	140000	166572	CMP	#140000,ANS1	;CHECK ANS1
012210	001401			BEQ	.+4	;BRANCH IF OK
012212	104002			HLT+2		;ANS1 NOT EQUAL TO 140000
012214	022767	000000	166562	CMP	#000000,ANS2	;CHECK ANS2
012222	001401			BEQ	.+4	;BRANCH IF OK
012224	104002			HLT+2		;ANS2 NOT EQUAL TO 000000
012226	174267	166554		STF	AC2, ANS3	;STORE AC2 IN ANS3, ANS4
012232	022767	040000	166546	CMP	#040000,ANS3	;CHECK ANS3
012240	001401			BEQ	.+4	;BRANCH IF OK
012242	104004			HLT+4		;AC2 CHANGED
012244	022767	000000	166536	CMP	#000000,ANS4	;CHECK ANS4
012252	001401			BEQ	.+4	;BRANCH IF OK
012254	104004			HLT+4		;AC2 CHANGED

```

:*****
:TEST 77:      TEST SUBF (SUBTRACT FLOATING)
:              077777,177777 - 177777,177777 = 000177,177777
:              FPS = 147406,   FSRC = MO-AC2,   AC = AC1
:              FEC = 10,     FEA = FPI77
:*****

```

012256	104400			SCOPE		
012260	000404			BR	TST77	;BRANCH OVER INPUT DATA

```

012262 077777 177777      DTA77: 077777,177777
012266 177777 177777      DTB77: 177777,177777

012272 170127 047417      TST77: LDFPS  #047417      ;LOAD FLOATING POINT STATUS
012276 172567 177760      LDF      DTA77,  AC1      ;LOAD 077777,177777 INTO AC1
012302 172667 177760      LDF      DTB77,  AC2      ;LOAD 177777,177777 INTO AC2
012306 173102                SUBF     AC2,  AC1      ;SUBTRACT AC2 FROM AC1
012310 170200                STFPS   FPS          ;STORE FLOATING POINT STATUS
012312 170367 165504      STST    FEC          ;STORE EXCEPTION CODES
012316 022700 147406      CMP     #147406,FPS    ;CHECK FLOATING POINT STATUS
012322 001401                BEQ     .+4           ;BRANCH IF OK
012324 104000                HLT                    ;FPS NOT EQUAL TO 147406

012326 022767 000010 166466      CMP     #10,  FEC      ;CHECK FLOATING EXCEPTION CODE
012334 001401                BEQ     .+4           ;BRANCH IF OK
012336 104000                HLT                    ;FEC NOT EQUAL TO 10

012340 174167 166436                STF     AC1,  ANS1      ;STORE DIFFERENCE IN ANS1, ANS2
012344 022767 000177 16643C      CMP     #000177,ANS1   ;CHECK ANS1
012352 001401                BEQ     .+4           ;BRANCH IF OK
012354 104002                HLT+2                ;ANS1 NOT EQUAL TO 000177

012356 022767 177777 166420      CMP     #177777,ANS2   ;CHECK ANS2
012364 001401                BEQ     .+4           ;BRANCH IF OK
012366 104002                HLT+2                ;ANS2 NOT EQUAL TO 177777

012370 174267 166412                STF     AC2,  ANS3      ;STORE AC2 IN ANS3, ANS4
012374 022767 177777 166404      CMP     #177777,ANS3   ;CHECK ANS3
012402 001401                BEQ     .+4           ;BRANCH IF OK
012404 104004                HLT+4                ;AC2 CHANGED

012406 022767 177777 166374      CMP     #177777,ANS4   ;CHECK ANS4
012414 001401                BEQ     .+4           ;BRANCH IF OK
012416 104004                HLT+4                ;AC2 CHANGED

*****
:TEST 100:      TEST SUBD (SUBTRACT DOUBLE PRECISION)
:              000000,000000,000000,000000 - 020000,000000,000000,000000 =
:              120000,000000,000000,000000
:              FPS = 047610,  FSRC = M6-R7,  AC = AC2
*****

012420 104400                SCOPE
012422 000410                BR      TST100        ;BRANCH OVER INPL* DATA

012424 000000 000000 000000  DTA100: 000000,000000,000000,000000
012432 000000                DTB100: 020000,000000,000000,000000
012434 020000 000000 000000
012442 000000

012444 170127 047617      TST100: LDFPS  #047617      ;LOAD FLOATING POINT STATUS
012450 172667 177760      DC      DTA100,  AC2    ;LOAD 000000,000000,000000,000000 INTO AC2
012454 173267 177764      SUBD   DTB100,  AC2    ;SUBTRACT 020000,000000,000000,000000 FROM AC2
012456 170200                STFPS   FPS          ;STORE FLOATING POINT STATUS

```

B05

FINANCE-11-20CPD-1
00000000

TEST OF ADDF. ADDD. SUBF. SUBD MACY11 27(732) 17-SEP-76 09.41 PAGE 53
TEST SECTION

012460	022700	047610		CMP	#047610,FPS	:CHECK FLOATING POINT STATUS
012462	001401			BEQ	.+4	:BRANCH IF OK
012464	104000			HLT		:FPS NOT EQUAL TO 047610
012472	174267	166304		STD	AC2 ANS1	:STORE DIFFERENCE IN ANS1 THRU ANS4
012474	022767	120000	166276	CMP	#120000,ANS1	:CHECK ANS1
012502	001401			BEQ	.+4	:BRANCH IF OK
012504	104004			HLT+4		:ANS1 NOT EQUAL TO 120000
012512	022767	000000	166266	CMP	#000000,ANS2	:CHECK ANS2
012514	001401			BEQ	.+4	:BRANCH IF OK
012516	104004			HLT+4		:ANS2 NOT EQUAL TO 000000
012522	022767	000000	166256	CMP	#000000,ANS3	:CHECK ANS3
012524	001401			BEQ	.+4	:BRANCH IF OK
012526	104004			HLT+4		:ANS3 NOT EQUAL TO 000000
012532	022767	000000	166246	CMP	#000000,ANS4	:CHECK ANS4
012534	001401			BEQ	.+4	:BRANCH IF OK
012536	104004			HLT+4		:ANS4 NOT EQUAL TO 000000

 TEST 101: TEST SUBD (SUBTRACT DOUBLE PRECISION)
 040177,177777,177777,177777 - 140177,177777,177777,177777 =
 040377,177777,177777,177777
 FPS = 047600, FSRC = M6-R7, AC = AC3

012546	104400			SCOPE		
012552	000410			BR	TEST101	:BRANCH OVER INPUT DATA
012558	040177	177777	177777	DATA01:	040177,177777,177777,177777	
012560	177777					
012562	140177	177777	177777	DATA01:	140177,177777,177777,177777	
012570	:77777					
012572	170127	047617		TEST01:	LDFPS #047617	:LOAD FLOATING POINT STATUS
012574	172767	177750			DATA01, AC3	:LOAD 040177,177777,177777,177777 INTO AC3
012576	173367	177754		DATA01:	DATA01, AC3	:SUBTRACT 140177,177777,177777,177777 FROM AC3
012578	175200			DATA01:	FPS	:STORE FLOATING POINT STATUS
012580	022700	047600			#047600,FPS	:CHECK FLOATING POINT STATUS
012582	001401				.+4	:BRANCH IF OK
012584	104000					:FPS NOT EQUAL TO 047600
012590	174367	166156		STD	AC3 ANS1	:STORE DIFFERENCE IN ANS1 THRU ANS4
012592	022767	040377	166150	CMP	#040377,ANS1	:CHECK ANS1
012594	001401			BEQ	.+4	:BRANCH IF OK
012596	104004			HLT+4		:ANS1 NOT EQUAL TO 040377
012602	022767	177777	166140	CMP	#177777,ANS2	:CHECK ANS2
012604	001401			BEQ	.+4	:BRANCH IF OK
012606	104004			HLT+4		:ANS2 NOT EQUAL TO 177777
012612	022767	177777	166130	CMP	#177777,ANS3	:CHECK ANS3
012614	001401			BEQ	.+4	:BRANCH IF OK

C05

MANNOEC-11-20FPC-1
00000000

TEST OF ADDF, ADDD, SUBF, SUBD MACY11 27(732) 17-SEP-76 09:41 PAGE 54
TEST SECTION

```

012660 104004          HLT+4          :ANS3 NOT EQUAL TO 177777
012662 022767 177777 166120      CMP          #177777,ANS4      :CHECK ANS4
012664 001401          BEQ          .+4          :BRANCH IF OK
012666 104004          HLT+4          :ANS4 NOT EQUAL TO 177777

```

```

*****
TEST 102:      TEST SUBD (SUBTRACT DOUBLE PRECISION)
              040125,052525,052525,052525 - 140052,125252,125252,125252 =
              040300,000000,000000,000000
              FPS = 047600,   FSRC = M6-R7,   AC = AC2
*****

```

```

012674 104400          SCOPE
012676 000410          BR          TEST102          :BRANCH OVER INPUT DATA

```

```

012700 040125 052525 052525 DTA102: 040125,052525,052525,052525
012706 052525
012710 140052 125252 125252 DTB102: 140052,125252,125252,125252
012716 125252

```

```

012720 170127 047617          TEST102: LD FPS          #047617          :LOAD FLOATING POINT STATUS
012724 172667 177750          LDD          DTA102, AC2          :LOAD 040125,052525,052525,052525 INTO AC2
012730 173267 177754          SUBJBC       DTB102, AC2          :SUBTRACT 140052,125252,125252,125252 FROM AC2
012734 170200          ST FPS          :STORE FLOATING POINT STATUS
012738 022700 047600          COMP        #047600, FPS          :CHECK FLOATING POINT STATUS
012742 001401          BEQ          .+4          :BRANCH IF OK
012744 104000          HLT+4          :FPS NOT EQUAL TO 047600

```

```

012746 174267 166030          STD          AC2, ANS1          :STORE DIFFERENCE IN ANS1 FROM ANS4
012750 022767 040300 166022      COMF        #040300, ANS1          :CHECK ANS1
012756 001401          BEQ          .+4          :BRANCH IF OK
012762 104004          HLT+4          :ANS1 NOT EQUAL TO 040300

```

```

012764 022767 000000 166012      CMP          #000000, ANS2          :CHECK ANS2
012770 001401          BEQ          .+4          :BRANCH IF OK
012774 104004          HLT+4          :ANS2 NOT EQUAL TO 000000

```

```

012776 022767 000000 166002      CMP          #000000, ANS3          :CHECK ANS3
013004 001401          BEQ          .+4          :BRANCH IF OK
013006 104004          HLT+4          :ANS3 NOT EQUAL TO 000000

```

```

013010 022767 000000 165772      CMP          #000000, ANS4          :CHECK ANS4
013016 001401          BEQ          .+4          :BRANCH IF OK
013022 104004          HLT+4          :ANS4 NOT EQUAL TO 000000

```

```

*****
TEST 103:      TEST SUBC (SUBTRACT DOUBLE PRECISION)
              040052,125252,125252,125252 - 140052,125252,125252,125252 =
              040252,125252,125252,125252
              FPS = 047600,   FSRC = M6-R7,   AC = AC0
*****

```

```

013022 104000          SCOPE

```

```

013024 000410 BR TST103 :BRANCH OVER INPUT DATA
013026 040052 125252 125252 DTA103: 040052,125252,125252,125252
013028 125252 125252 DTB103: 140052,125252,125252,125252
013030 170127 047617 TST103: LDFPS #047617 :LOAD FLOATING POINT STATUS
013032 177750 LDD DTA103, ACC :LOAD 040052,125252,125252,125252 INTO ACC
013034 177754 FP1:03: SUBD DTB103, ACC :SUBTRACT 140052,125252,125252,125252 FROM ACC
013036 170200 STFPS FPS :STORE FLOATING POINT STATUS
013038 022700 047600 CMP #047600, FPS :CHECK FLOATING POINT STATUS
013040 001401 BEQ .+4 :BRANCH IF OK
013042 104000 F .+4 :FPS NOT EQUAL TO 047600

013074 174067 165702 STD ACC ANS1 :STORE DIFFERENCE IN ANS1 FROM ANS4
013076 022767 040252 165674 SUBC #040252, ANS1 :CHECK ANS1
013078 001401 BEQ .+4 :BRANCH IF OK
013080 104004 F .+4 :ANS1 NOT EQUAL TO 040252

013112 022767 125252 165664 BNEF #125252, ANS2 :CHECK ANS2
013114 001401 BEQ .+4 :BRANCH IF OK
013116 104004 F .+4 :ANS2 NOT EQUAL TO 125252

013122 022767 125252 165654 CMP #125252, ANS3 :CHECK ANS3
013124 001401 BEQ .+4 :BRANCH IF OK
013126 104004 F .+4 :ANS3 NOT EQUAL TO 125252

013136 022767 125252 165644 CMP #125252, ANS4 :CHECK ANS4
013138 001401 BEQ .+4 :BRANCH IF OK
013140 104004 F .+4 :ANS4 NOT EQUAL TO 125252

```

```

*****
TEST 104: TEST SUBD (SUBTRACT DOUBLE PRECISION)
077777,177777,177777,177777 - 077777,177777,177777,177777 =
000000,000000,000000,000000
FPS = 047604, FSRC = M6-R7, AC = AC2
*****

```

```

013150 104100 SCOPE
013152 000410 BR TST104 :BRANCH OVER INPUT DATA
013154 077777 177777 177777 DTA104: 077777,177777,177777,177777
013156 177777 177777 177777 DTB104: 077777,177777,177777,177777
013158 177777
013172 177777

013174 170127 047617 TST104: LDFPS #047617 :LOAD FLOATING POINT STATUS
013176 177750 LDD DTA104, ACC :LOAD 077777,177777,177777,177777 INTO ACC
013178 177754 FP1:04: SUBD DTB104, ACC :SUBTRACT 077777,177777,177777,177777 FROM ACC
013180 170200 STFPS FPS :STORE FLOATING POINT STATUS
013182 047604 047604 CMP #047604, FPS :CHECK FLOATING POINT STATUS
013184 001401 BEQ .+4 :BRANCH IF OK
013186 104100 F .+4 :FPS NOT EQUAL TO 047604

```


00000000-00000000
00000000

013222	174267	165554		STO	AC2, ANS1	:STORE DIFFERENCE IN ANS1 THRU ANS4
013226	022767	000000	165546	CMP	#000000,ANS1	:CHECK ANS1
013234	001401			BEQ	+.4	:BRANCH IF OK
013236	104004			HLT+4		:ANS1 NOT EQUAL TO 000000
013240	022767	000000	165536	CMP	#000000,ANS2	:CHECK ANS2
013246	001401			BEQ	+.4	:BRANCH IF OK
013250	104004			HLT+4		:ANS2 NOT EQUAL TO 000000
013252	022767	000000	165526	CMP	#000000,ANS3	:CHECK ANS3
013256	001401			BEQ	+.4	:BRANCH IF OK
013260	104004			HLT+4		:ANS3 NOT EQUAL TO 000000
013264	022767	000000	165516	CMP	#000000,ANS4	:CHECK ANS4
013270	001401			BEQ	+.4	:BRANCH IF OK
013274	104004			HLT+4		:ANS4 NOT EQUAL TO 000000

 TEST 105: TEST SUBD (SUBTRACT DOUBLE PRECISION)
 040252,125252,125252,125252 - 137525,052525,052525,052525 =
 040305,052525,052525,052525
 FPS = 047600, FSRC = M6-R7, AC = AC1

013276	104400			SCOPE		
013300	000410			BR	TST105	:BRANCH OVER INPUT DATA
013302	040252	125252	125252	DTA105:	040252,125252,125252,125252	
013310	125252					
013312	137525	052525	052525	DTB105:	137525,052525,052525,052525	
013320	052525					
013322	170127	047617		TST105:	LDFPS #047617	:LOAD FLOATING POINT STATUS
013326	172567	177750		LDD	DTA105, AC1	:LOAD 040252,125252,125252,125252 INTO AC1
013332	173167	177754		FP105:	SUBD DTB105, AC1	:SUBTRACT 137525,052525,052525,052525 FROM AC1
013336	170200			STFPS	FPS	:STORE FLOATING POINT STATUS
013340	022700	047600		CMP	#047600,FPS	:CHECK FLOATING POINT STATUS
013344	001401			BEQ	+.4	:BRANCH IF OK
013346	104000			HLT		:FPS NOT EQUAL TO 047600
013350	174167	165426		STO	AC1, ANS1	:STORE DIFFERENCE IN ANS1 THRU ANS4
013354	022767	040305	165420	CMP	#040305,ANS1	:CHECK ANS1
013362	001401			BEQ	+.4	:BRANCH IF OK
013364	104004			HLT+4		:ANS1 NOT EQUAL TO 040305
013366	022767	052525	165410	CMP	#052525,ANS2	:CHECK ANS2
013374	001401			BEQ	+.4	:BRANCH IF OK
013376	104004			HLT+4		:ANS2 NOT EQUAL TO 052525
013380	022767	052525	165400	CMP	#052525,ANS3	:CHECK ANS3
013386	001401			BEQ	+.4	:BRANCH IF OK
013390	104004			HLT+4		:ANS3 NOT EQUAL TO 052525
013392	022767	052525	165370	CMP	#052525,ANS4	:CHECK ANS4
013396	001401			BEQ	+.4	:BRANCH IF OK

F05

MAINP... 1-20CPD-C
TEST SECTION

TEST OF ADDF, ADDD, SUBF, SUBD MACY11 27(732) 17-SEP-76 09:41 PAGE 57
TEST SECTION

013422 104004

HLT+4

;ANS4 NOT EQUAL TO 052525

TEST 106: TEST SUBD (SUBTRACT DOUBLE PRECISION)
137525,052525,052525,052525 - 040252,125252,125252,125252 =
140305,052525,052525,052525
FPS = 047610, FSRC = M6-R7, AC = ACC

013422 104400
013422 000410

SCOPE

BR TST106

;BRANCH OVER INPUT DATA

013430 137525
013436 052525
013440 040252
013446 125252

052525 052525
125252 125252

DTA106: 137525,052525,052525,052525
DTB106: 040252,125252,125252,125252

013450 170127
013454 172467
013460 173067
013464 170200
013466 022700
013472 001401
013474 104000

047617 177750
177754 047610

TST106: LDFPS #047617
LDD DTA106, ACC
SUBD DTB106, ACC
STFPS FPS
CMP #047610, FPS
BEQ .+4
HLT

;LOAD FLOATING POINT STATUS
;LOAD 137525,052525,052525,052525 INTO ACC
;SUBTRACT 040252,125252,125252,125252 FROM ACC
;STORE FLOATING POINT STATUS
;CHECK FLOATING POINT STATUS
;BRANCH IF OK
;FPS NOT EQUAL TO 047610

013476 174067
013502 022767
013510 001401
013512 104004

165300 140305
165272

STC ACC ANS1
CMP #140305, ANS1
BEQ .+4
HLT+4

;STORE DIFFERENCE IN ANS1 FROM ANS4
;CHECK ANS1
;BRANCH IF OK
;ANS1 NOT EQUAL TO 140305

013514 022767
013522 001401
013524 104004

052525 165262

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

;CHECK ANS2
;BRANCH IF OK
;ANS2 NOT EQUAL TO 052525

013526 022767
013534 001401
013536 104004

052525 165252

CMP #052525, ANS3
BEQ .+4
HLT+4

;CHECK ANS3
;BRANCH IF OK
;ANS3 NOT EQUAL TO 052525

013538 022767
013546 001401
013548 104004

052525 165242

CMP #052525, ANS4
BEQ .+4
HLT+4

;CHECK ANS4
;BRANCH IF OK
;ANS4 NOT EQUAL TO 052525

TEST 107: TEST SUBD (SUBTRACT DOUBLE PRECISION)
077777,177777,177777,177777 - 100200,000000,000000,000000 =
077777,177777,177777,177777
FPS = 047600, FSRC = M6-R7, AC = ACC

013552 104400
013554 000410

SCOPE

BR TST107

;BRANCH OVER INPUT DATA

013556 077777
013558 177777

177777 177777

DTA107: 077777,177777,177777,177777

G05

MAINTENANCE - 205FD-C
00000000

TEST OF ADD, ADD, SUB, SUBC MACY11 27(732) 17-SEP-76 09:41 PAGE 58
TEST SECTION

```

013566 100200 000000 000000 DTB107: 100200,000000,000000,000000
013574 000000

013576 170127 047617 TST107: LDFPS #047617 :LOAD FLOATING POINT STATUS
013602 172567 177750 :LDD DTB107, AC1 :LOAD 077777,177777,177777,177777 INTO AC1
013606 173167 177754 :SUBC DTB107, AC1 :SUBTRACT 100200,000000,000000,000000 FROM AC1
013612 170200 :STFPS FPS :STORE FLOATING POINT STATUS
013614 022700 047600 :CMP #047600,FPS :CHECK FLOATING POINT STATUS
013620 001401 :BEQ .+4 :BRANCH IF OK
013622 104000 :HLT :FPS NOT EQUAL TO 047600

013624 174167 165152 :STD AC1,ANS1 :STORE DIFFERENCE IN ANS1 *4, ANS1
013630 022767 077777 165144 :LMP #077777,ANS1 :CHECK ANS1
013636 001401 :BEQ .+4 :BRANCH IF OK
013640 104004 :HLT :ANS1 NOT EQUAL TO 077777

013642 022767 177777 165134 :CMP #177777,ANS2 :CHECK ANS2
013650 001401 :BEQ .+4 :BRANCH IF OK
013652 104004 :HLT :ANS2 NOT EQUAL TO 177777

013654 022767 177777 165124 :CMP #177777,ANS3 :CHECK ANS3
013662 001401 :BEQ .+4 :BRANCH IF OK
013664 104004 :HLT :ANS3 NOT EQUAL TO 177777

013666 022767 177777 165114 :CMP #177777,ANS4 :CHECK ANS4
013674 001401 :BEQ .+4 :BRANCH IF OK
013676 104004 :HLT :ANS4 NOT EQUAL TO 177777

```

```

*****
TEST 110: TEST SUBC (SUBTRACT DOUBLE PRECISION)
          000200,000000,000000,000000 = 077777,177777,177777,177777
          177777,177777,177777,177777
          FPS = 047610, FSRC = M6-R7, AC = ACC
*****

```

```

013700 104000 :BR BR :BRANCH OVER INPUT DATA
013702 000400

013704 000200 000000 000000 DTB110: 000200,000000,000000,000000
013710 000000
013712 077777 177777 DTB110: 077777,177777,177777,177777
013714 077777
013722 177777

013724 170127 047617 *ST110: LDFPS #047617 :LOAD FLOATING POINT STATUS
013730 172467 177750 :LDD DTB110, ACC :LOAD 000200,000000,000000,000000 INTO ACC
013734 173067 177754 :SUBC DTB110, ACC :SUBTRACT 077777,177777,177777,177777 FROM ACC
013740 170200 :STFPS FPS :STORE FLOATING POINT STATUS
013746 022700 047610 :CMP #047610,FPS :CHECK FLOATING POINT STATUS
013750 001401 :BEQ .+4 :BRANCH IF OK
013752 104000 :HLT :FPS NOT EQUAL TO 047610

013754 174167 165024 :STD ACC,ANS1 :STORE DIFFERENCE IN ANS1 *4, ANS1
013760 022767 177777 165016 :LMP #177777,ANS1 :CHECK ANS1
013766 001401 :BEQ .+4 :BRANCH IF OK
013770 104004 :HLT :ANS1 NOT EQUAL TO 177777

```

H05

MAINTENANCE - 1-00FPO-0
COPYRIGHT

TEST OF ADD, ADD, SUB, SUBC
TEST SECTION

MACY11 27(732) 17-SEP-76 09:41 PAGE 59

014026	022767	177777	165006	CMP	#177777,ANS2	:CHECK ANS2
014027	001401			BEG	..+4	:BRANCH IF OK
014028	104004			HLT+4		:ANS2 NOT EQUAL TO 177777
014029	022767	177777	164776	CMP	#177777,ANS3	:CHECK ANS3
014030	001401			BEG	..+4	:BRANCH IF OK
014031	104004			HLT+4		:ANS3 NOT EQUAL TO 177777
014032	022767	177777	164766	CMP	#177777,ANS4	:CHECK ANS4
014033	001401			BEG	..+4	:BRANCH IF OK
014034	104004			HLT+4		:ANS4 NOT EQUAL TO 177777

 TEST 111: TEST SUBC SUBTRACT DOUBLE PRECISION)
 03:152,125252,125252,125252 - 147125,052525,052525,052525 =
 047125,052525,052525,052525
 FPS = 047600, FSRC = ME-P7, AC = AC1

014026	104400			SCOPE		
014030	000410			BR	*S*111	:BRANCH OVER INPUT DATA
014032	031152	125252	125252	DTA111:	031152,125252,125252,125252	
014040	125252					
014042	147125	052525	052525	DTB111:	147125,052525,052525,052525	
014050	052525					

014052	170127	047617		TST111:	LDFPS	#047617	:LOAD FLOATING POINT STATUS
014056	172567	177750			LDAC	DTA111, AC1	:LOAD 031152,125252,125252,125252 INTO AC1
014062	173167	177754		FP111:	SUBC	DTB111, AC1	:SUBTRACT 147125,052525,052525,052525 FROM AC1
014066	170200				STFPS	FPS	:STORE FLOATING POINT STATUS
014070	022700	047600			CMP	#047600,FPS	:CHECK FLOATING POINT STATUS
014074	001401				BEG	..+4	:BRANCH IF OK
014076	104000				HLT		:FPS NOT EQUAL TO 047600

014100	174167	164676		STC	AC1, ANS1	:STORE DIFFERENCE IN ANS1 THE, ANS4
014104	022767	047125	164670	CMP	#047125,ANS1	:CHECK ANS1
014108	001401			BEG	..+4	:BRANCH IF OK
014114	104004			IF+4		:ANS1 NOT EQUAL TO 047125

014116	022767	052525	164660	CMP	#052525,ANS2	:CHECK ANS2
014120	001401			BEG	..+4	:BRANCH IF OK
014126	104004			IF+4		:ANS2 NOT EQUAL TO 052525

014130	022767	052525	164650	CMP	#052525,ANS3	:CHECK ANS3
014134	001401			BEG	..+4	:BRANCH IF OK
014140	104004			IF+4		:ANS3 NOT EQUAL TO 052525

014142	022767	052526	164640	CMP	#052526,ANS4	:CHECK ANS4
014146	001401			BEG	..+4	:BRANCH IF OK
014152	104004			IF+4		:ANS4 NOT EQUAL TO 052526

TEST 112: TEST SUBD (SUBTRACT DOUBLE PRECISION)
031152,125252,125252,125252 - 147325,052525,052525,052525 =
047325,052525,052525,052525
FPS = 047600, FSRC = M6-R7, AC = AC3

014154 104400
014156 000410

SCOPE
BR TST112 :BRANCH OVER INPUT DATA

014160 031152 125252 125252
014166 125252
014170 147325 052525 052525
014176 052525

DTA112: 031152,125252,125252,125252
DTB112: 147325,052525,052525,052525

014200 170127 047617
014204 172767 177750
014210 173367 177754
014214 170200
014216 022700 047600
014222 001401
014224 104000

TST112: LDFPS #047617 :LOAD FLOATING POINT STATUS
LDD DTA112, AC3 :LOAD 031152,125252,125252,125252 INTO AC3
FP112: SUBD DTB112, AC3 :SUBTRACT 147325,052525,052525,052525 FROM AC3
STFPS FPS :STORE FLOATING POINT STATUS
CMP #047600,FPS :CHECK FLOATING POINT STATUS
BEQ .+4 :BRANCH IF OK
HLT :FPS NOT EQUAL TO 047600

014226 174367 164550
014232 022767 047325 164542
014240 001401
014242 104004

STD AC3, ANS1 :STORE DIFFERENCE IN ANS1 THRU ANS4
CMP #047325,ANS1 :CHECK ANS1
BEQ .+4 :BRANCH IF OK
HLT+4 :ANS1 NOT EQUAL TO 047325

014244 022767 052525 164532
014252 001401
014254 104004

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

014256 022767 052525 164522
014264 001401
014266 104004

CMP #052525,ANS3 :CHECK ANS3
BEQ .+4 :BRANCH IF OK
HLT+4 :ANS3 NOT EQUAL TO 052525

014270 022767 052525 164512
014276 001401
014300 104004

CMP #052525,ANS4 :CHECK ANS4
BEQ .+4 :BRANCH IF OK
HLT+4 :ANS4 NOT EQUAL TO 052525

TEST 113: TEST SUBD (SUBTRACT DOUBLE PRECISION)
031152,125252,125252,125252 - 047125,052525,052525,052525 =
147125,052525,052525,052524
FPS = 047610, FSRC = M6-R7, AC = AC0

014302 104400
014304 000410

SCOPE
BR TST113 :BRANCH OVER INPUT DATA

014306 031152 125252 125252
014314 125252
014316 047125 052525 052525
014324 052525

DTA113: 031152,125252,125252,125252
DTB113: 047125,052525,052525,052525

014326 170127 047617

TST113: LDFPS #047617 :LOAD FLOATING POINT STATUS

014332	172467	177750		LD0	DTA113, ACO	;LOAD 031152,125252,125252,125252 INTO ACO
014336	173067	177754		FP1113: SUBD	DTB113, ACO	;SUBTRACT 047125,052525,052525,052525 FROM ACO
014342	170200			STFPS	FPS	;STORE FLOATING POINT STATUS
014344	022700	047610		CMP	#047610,FPS	;CHECK FLOATING POINT STATUS
014350	001401			BEQ	+.4	;BRANCH IF OK
014352	104000			HLT		;FPS NOT EQUAL TO 047610
014354	174067	164422		STD	ACO, ANS1	;STORE DIFFERENCE IN ANS1 THPL ANS4
014360	022767	147125	164414	CMP	#147125,ANS1	;CHECK ANS1
014366	001401			BEQ	+.4	;BRANCH IF OK
014370	104004			HLT+4		;ANS1 NOT EQUAL TO 147125
014372	022767	052525	164424	CMP	#052525,ANS2	;CHECK ANS2
014400	001401			BEQ	+.4	;BRANCH IF OK
014402	104004			HLT+4		;ANS2 NOT EQUAL TO 052525
014404	022767	052525	164374	CMP	#052525,ANS3	;CHECK ANS3
014412	001401			BEQ	+.4	;BRANCH IF OK
014414	104004			HLT+4		;ANS3 NOT EQUAL TO 052525
014416	022767	052524	164364	CMP	#052524,ANS4	;CHECK ANS4
014424	001401			BEQ	+.4	;BRANCH IF OK
014426	104004			HLT+4		;ANS4 NOT EQUAL TO 052524

```

*****
:TEST 114: TEST SUBD (SUBTRACT DOUBLE PERCISION)
:          077652,125252,125252,125252 - 177452,125252,125252,125252 =
:          077777,177777,177777,177777
:          FPS = 047600, FSRC = M6-R7, AC = AC3
*****

```

014430	104400			SCOPE		
014432	000410			BR	TST114	;BRANCH OVER INPUT DATA
014434	077652	125252	125252	DTA114:	077652,125252,125252,125252	
014442	125252					
014444	177452	125252	125252	DTB114:	177452,125252,125252,125252	
014452	125252					
014454	170127	047617		TST114: LDFPS	#047617	;LOAD FLOATING POINT STATUS
014460	172767	177750		LD0	DTA114, AC3	;LOAD 077652,125252,125252,125252 INTO AC3
014464	173367	177754		FP1114: SUBD	DTB114, AC3	;SUBTRACT 177452,125252,125252,125252 FROM AC3
014470	170200			STFPS	FPS	;STORE FLOATING POINT STATUS
014472	022700	047600		CMP	#047600,FPS	;CHECK FLOATING POINT STATUS
014476	001401			BEQ	+.4	;BRANCH IF OK
014500	104000			HLT		;FPS NOT EQUAL TO 047600
014502	174367	164274		STD	AC3, ANS1	;STORE DIFFERENCE IN ANS1 THPL ANS4
014506	022767	077777	164266	CMP	#077777,ANS1	;CHECK ANS1
014514	001401			BEQ	+.4	;BRANCH IF OK
014516	104004			HLT+4		;ANS1 NOT EQUAL TO 077777
014520	022767	177777	164256	CMP	#177777,ANS2	;CHECK ANS2
014526	001401			BEQ	+.4	;BRANCH IF OK
014530	104004			HLT+4		;ANS2 NOT EQUAL TO 177777

```

014532 022767 177777 164246      CMP      #177777,ANS3      ;CHECK ANS3
014540 001401      BEQ      .+4              ;BRANCH IF OK
014542 104004      HLT+4          ;ANS3 NOT EQUAL TO 177777

014544 022767 177777 164236      CMP      #177777,ANS4      ;CHECK ANS4
014552 001401      BEQ      .+4              ;BRANCH IF OK
014554 104004      HLT+4          ;ANS4 NOT EQUAL TO 177777

```

```

:*****
:TEST 115:      TEST SUBD (SUBTRACT DOUBLE PERCISION)
:              177452,125252,125252,125252 - 077652,125252,125252,125252 =
:              177777,177777,177777,177777
:              FPS = 047610,   FSRC = M6-R7,   AC = ACO
:*****

```

```

014556 104400      SCOPE
014560 000410      BR      TST115          ;BRANCH OVER INPUT DATA

014562 177452 125252 125252 DTA115: 177452,125252,125252,125252
014570 125252
014572 077652 125252 125252 DTB115: 077652,125252,125252,125252
014600 125252

014602 170127 047617      TST115: LDFPS      #047617      ;LOAD FLOATING POINT STATUS
014606 172467 177750      LDD      DTA115, ACC      ;LOAD 177452,125252,125252,125252 INTO ACC
014612 173067 177754      FPI115: SUBD     DTB115, ACC ;SUBTRACT 077652,125252,125252,125252 FROM ACC
014616 170200      STFPS     FPS              ;STORE FLOATING POINT STATUS
014620 022700 047610      CMP      #047610,FPS      ;CHECK FLOATING POINT STATUS
014624 001401      BEQ      .+4              ;BRANCH IF OK
014626 104000      HLT          ;FPS NOT EQUAL TO 047610

014630 174067 164146      STC      ACC, ANS1        ;STORE DIFFERENCE IN ANS1 THRU ANS4
014634 022767 177777 164140      CMP      #177777,ANS1     ;CHECK ANS1
014642 001401      BEQ      .+4              ;BRANCH IF OK
014644 104004      HLT+4          ;ANS1 NOT EQUAL TO 177777

014646 022767 177777 164130      CMP      #177777,ANS2     ;CHECK ANS2
014654 001401      BEQ      .+4              ;BRANCH IF OK
014656 104004      HLT+4          ;ANS2 NOT EQUAL TO 177777

014660 022767 177777 164120      CMP      #177777,ANS3     ;CHECK ANS3
014666 001401      BEQ      .+4              ;BRANCH IF OK
014670 104004      HLT+4          ;ANS3 NOT EQUAL TO 177777

014672 022767 177777 164110      CMP      #177777,ANS4     ;CHECK ANS4
014700 001401      BEQ      .+4              ;BRANCH IF OK
014702 104004      HLT+4          ;ANS4 NOT EQUAL TO 177777

```

```

:*****
:TEST 116:      TEST SUBD (SUBTRACT DOUBLE PERCISION)
:              000425,052525,052525,052525 - 000252,125252,125252,125252 =
:              000200,000000,000000,000000
:              FPS = 047600,   FSRC = M6-R7,   AC = A02
:*****

```

```

014704 104400          SCOPE
014706 000410          BR      TST116          ;BRANCH OVER INPUT DATA

014710 000425 052525 052525 DTA116: 000425,052525,052525,052525
014716 052525
014720 000252 125252 125252 DTB116: 000252,125252,125252,125252
014726 125252

014730 170127 047617      TST116: LDFPS      #047617          ;LOAD FLOATING POINT STATUS
014734 172667 177750      LDD      DTA116, AC2          ;LOAD 000425,052525,052525,052525 INTO AC2
014740 173267 177754      FPI116: SUBD      DTB116, AC2          ;SUBTRACT 000252,125252,125252,125252 FROM AC2
014744 170200      STFPS      FPS          ;STORE FLOATING POINT STATUS
014746 022700 047600      CMP      #047600,FPS          ;CHECK FLOATING POINT STATUS
014752 001401      BEQ      .+4          ;BRANCH IF OK
014754 104000      HLT          ;FPS NOT EQUAL TO 047600

014756 174267 164020      STD      AC2, ANS1          ;STORE DIFFERENCE IN ANS1 THRU ANS4
014762 022767 000200 164012      CMP      #000200,ANS1          ;CHECK ANS1
014770 001401      BEQ      .+4          ;BRANCH IF OK
014772 104004      HLT+4          ;ANS1 NOT EQUAL TO 000200

014774 022767 000000 164002      CMP      #000000,ANS2          ;CHECK ANS2
015002 001401      BEQ      .+4          ;BRANCH IF OK
015004 104004      HLT+4          ;ANS2 NOT EQUAL TO 000000

015006 022767 000000 163772      CMP      #000000,ANS3          ;CHECK ANS3
015014 001401      BEQ      .+4          ;BRANCH IF OK
015016 104004      HLT+4          ;ANS3 NOT EQUAL TO 000000

015020 022767 000000 163762      CMP      #000000,ANS4          ;CHECK ANS4
015026 001401      BEQ      .+4          ;BRANCH IF OK
015030 104004      HLT+4          ;ANS4 NOT EQUAL TO 000000

```

```

*****
:TEST 117: TEST SUBD (SUBTRACT DOUBLE PRECISION)
:          100252,125252,125252,125252 - 100425,052525,052525,052525 =
:          000200,000000,000000,000000
:          FPS = 047600, FSRC = M6-R7, AC = AC2
*****

```

```

015032 104400          SCOPE
015034 000410          BR      TST117          ;BRANCH OVER INPUT DATA

015036 100252 125252 125252 DTA117: 100252,125252,125252,125252
015044 125252
015046 100425 052525 052525 DTB117: 100425,052525,052525,052525
015054 052525

015056 170127 047617      TST117: LDFPS      #047617          ;LOAD FLOATING POINT STATUS
015062 172667 177750      LDD      DTA117, AC2          ;LOAD 100252,125252,125252,125252 INTO AC2
015066 173267 177754      FPI117: SUBD      DTB117, AC2          ;SUBTRACT 100425,052525,052525,052525 FROM AC2
015072 170200      STFPS      FPS          ;STORE FLOATING POINT STATUS
015074 022700 047600      CMP      #047600,FPS          ;CHECK FLOATING POINT STATUS

```



```

015100 001401      BEQ      .+4      ;BRANCH IF OK
015102 104000      HLT              ;FPS NOT EQUAL TO 047600

015104 174267 163672 STD      AC2,   ANS1      ;STORE DIFFERENCE IN ANS1 THRU ANS4
015110 022767 000200 163664  CMP      #000200,ANS1    ;CHECK ANS1
015116 001401      BEQ      .+4      ;BRANCH IF OK
015120 104004      HLT+4      ;ANS1 NOT EQUAL TO 000200

015122 022767 000000 163654  CMP      #000000,ANS2    ;CHECK ANS2
015130 001401      BEQ      .+4      ;BRANCH IF OK
015132 104004      HLT+4      ;ANS2 NOT EQUAL TO 000000

015134 022767 000000 163644  CMP      #000000,ANS3    ;CHECK ANS3
015142 001401      BEQ      .+4      ;BRANCH IF OK
015144 104004      HLT+4      ;ANS3 NOT EQUAL TO 000000

015146 022767 000000 163634  CMP      #000000,ANS4    ;CHECK ANS4
015154 001401      BEQ      .+4      ;BRANCH IF OK
015156 104004      HLT+4      ;ANS4 NOT EQUAL TO 000000

```

```

*****
:TEST 120:      TEST SUBD (SUBTRACT DOUBLE PRECISION)
:              100177,177777,177777,177777 - 001252,125252,125252,125252 =
:              101252,125252,125252,125252
:              FPS = 047610,   FSRC = M6-R7,   AC = AC3
*****

```

```

015160 104400      SCOPE
015162 000410      BR      TST120      ;BRANCH OVER INPUT DATA

015164 100177 177777 177777 DTA120: 100177,177777,177777,177777
015172 177777
015174 001252 125252 125252 DTB120: 001252,125252,125252,125252
015202 125252

015204 170127 040200 TST120: LDFPS  #040200      ;LOAD FLOATING POINT STATUS
015210 172767 177750  LD      DTA120, AC3      ;LOAD 100177,177777,177777,177777 INTO AC3
015214 170127 047617  LDFPS  #047617      ;LOAD FLOATING POINT STATUS
015220 173367 177750  FPI120: SUBD  DTB120, AC3      ;SUBTRACT 001252,125252,125252,125252 FROM AC3
015224 170200  STFPS  FPS              ;STORE FLOATING POINT STATUS
015226 022700 047610  CMP      #047610,FPS      ;CHECK FLOATING POINT STATUS
015232 001401  BEQ      .+4      ;BRANCH IF OK
015234 104000  HLT              ;FPS NOT EQUAL TO 047610

015236 174367 163540 STD      AC3,   ANS1      ;STORE DIFFERENCE IN ANS1 THRU ANS4
015242 022767 101252 163532  CMP      #101252,ANS1    ;CHECK ANS1
015250 001401  BEQ      .+4      ;BRANCH IF OK
015252 104004  HLT+4      ;ANS1 NOT EQUAL TO 101252

015254 022767 125252 163522  CMP      #125252,ANS2    ;CHECK ANS2
015262 001401  BEQ      .+4      ;BRANCH IF OK
015264 104004  HLT+4      ;ANS2 NOT EQUAL TO 125252

015266 022767 125252 163512  CMP      #125252,ANS3    ;CHECK ANS3
015274 001401  BEQ      .+4      ;BRANCH IF OK

```

```

015276 104004          HLT+4          ;ANS3 NOT EQUAL TO 125252
015300 022767 125252 163502      CMP          #125252,ANS4      ;CHECK ANS4
015306 001401          BEQ          .+4            ;BRANCH IF OK
015310 104004          HLT+4          ;ANS4 NOT EQUAL TO 125252

```

```

:*****
:TEST 121:      TEST SUBD (SUBTRACT DOUBLE PRECISION)
:              100177.177777,177777,177777 - 100125.052525,052525,052525 =
:              100177,177777,177777,177777
:              FPS = 147617,   FSRC = M6-R7,   AC = AC3
:              FEC = 14,     FEA = FPI121
:*****

```

```

015312 104400          SCOPE
015314 000410          BR          TST121          ;BRANCH OVER INPJT DATA

```

```

015316 100177 177777 177777 DTA121: 100177,177777,177777,177777
015324 177777
015326 100125 052525 052525 DTB121: 100125,052525,052525,052525
015334 052525

```

```

015336 170127 040200      TST121: LDFPS          #040200          ;LOAD FLOATING POINT STATUS
015342 172767 177750      LDD          DTA121, AC3      ;LOAD 100177,177777,177777,177777 INTO AC3
015346 170127 047617      LDFPS          #047617          ;LOAD FLOATING POINT STATUS
015352 173367 177750      FPI121: SUBD          DTB121, AC3      ;SUBTRACT 100125,052525,052525,052525 FROM AC3
015356 170200          STFPS          FPS            ;STORE FLOATING POINT STATUS
015360 170367 163436      STST          FEC            ;STORE EXCEPTION CODES
015364 022700 147617      CMP          #147617,FPS      ;CHECK FLOATING POINT STATUS
015370 001401          BEQ          .+4            ;BRANCH IF OK
015372 104000          HLT

```

```

015374 022767 000014 163420      CMP          #14,   FEC        ;CHECK FLOATING EXCEPTION CODE
015402 001401          BEQ          .+4            ;BRANCH IF OK
015404 104000          HLT

```

```

015406 022767 015352 163410      CMP          #FPI121, FEA      ;CHECK FLOATING EXCEPTION ADDRESS
015414 001401          BEQ          .+4            ;BRANCH IF OK
015416 104000          HLT

```

```

015420 174367 163356          STD          AC3,   ANS1        ;STORE DIFFERENCE IN ANS1 THRU ANS4
015424 022767 100177 163350      CMP          #100177,ANS1      ;CHECK ANS1
015432 001401          BEQ          .+4            ;BRANCH IF OK
015434 104004          HLT+4          ;ANS1 NOT EQUAL TO 100177

```

```

015436 022767 177777 163340      CMP          #177777,ANS2      ;CHECK ANS2
015444 001401          BEQ          .+4            ;BRANCH IF OK
015446 104004          HLT+4          ;ANS2 NOT EQUAL TO 177777

```

```

015450 022767 177777 163330      CMP          #177777,ANS3      ;CHECK ANS3
015456 001401          BEQ          .+4            ;BRANCH IF OK
015460 104004          HLT+4          ;ANS3 NOT EQUAL TO 177777

```

```

015462 022767 177777 163320      CMP          #177777,ANS4      ;CHECK ANS4
015470 001401          BEQ          .+4            ;BRANCH IF OK

```

015472 104004

HLT+4

;ANS4 NOT EQUAL TO 177777

```

*****
TEST 122: TEST SUBD (SUBTRACT DOUBLE PRECISION)
          177652,125252,125252,125252 - 077452,125252,125252,125253 =
          100000,000000,000000,000000
          FPS = 046616, FSRC = M6-R7, AC = AC1
*****

```

015474 104400
015476 000410

SCOPE
BR

TST122

;BRANCH OVER INPUT DATA

015500 177652 125252 125252
015506 125252
015510 077452 125252 125252
015516 125253

DTA122: 177652,125252,125252,125252
DTB122: 077452,125252,125252,125253

015520 170127 046617
015524 172567 177750
015530 173167 177754
015534 170200
015536 022700 046616
015542 001401
015544 104000

TST122: LDFPS #046617 ;LOAD FLOATING POINT STATUS
LDD DTA122, AC1 ;LOAD 177652,125252,125252,125252 INTO AC1
FP122: SUBD DTB122, AC1 ;SUBTRACT 077452,125252,125252,125253 FROM AC1
STFPS FPS ;STORE FLOATING POINT STATUS
CMP #046616, FPS ;CHECK FLOATING POINT STATUS
BEQ .+4 ;BRANCH IF OK
HLT ;FPS NOT EQUAL TO 046616

015546 174167 163230
015552 022767 100000 163222
015560 001401
015562 104004

STC AC1, ANS1 ;STORE DIFFERENCE IN ANS1 THRU ANS4
CMP #100000, ANS1 ;CHECK ANS1
BEQ .+4 ;BRANCH IF OK
HLT+4 ;ANS1 NOT EQUAL TO 100000

;STORE DIFFERENCE IN ANS1 THRU ANS4
;CHECK ANS1
;BRANCH IF OK
;ANS1 NOT EQUAL TO 100000

015564 022767 000000 163212
015572 001401
015574 104004

CMP #000000, ANS2 ;CHECK ANS2
BEQ .+4 ;BRANCH IF OK
HLT+4 ;ANS2 NOT EQUAL TO 000000

;CHECK ANS2
;BRANCH IF OK
;ANS2 NOT EQUAL TO 000000

015576 022767 000000 163202
015584 001401
015586 104004

CMP #000000, ANS3 ;CHECK ANS3
BEQ .+4 ;BRANCH IF OK
HLT+4 ;ANS3 NOT EQUAL TO 000000

;CHECK ANS3
;BRANCH IF OK
;ANS3 NOT EQUAL TO 000000

015590 022767 000000 163172
015596 001401
015598 104004

CMP #000000, ANS4 ;CHECK ANS4
BEQ .+4 ;BRANCH IF OK
HLT+4 ;ANS4 NOT EQUAL TO 000000

;CHECK ANS4
;BRANCH IF OK
;ANS4 NOT EQUAL TO 000000

```

*****
TEST 123: TEST SUBD (SUBTRACT DOUBLE PRECISION)
          000252,125252,125252,125253 - 000425,052525,052525,052525 =
          000000,000000,000000,000000
          FPS = 045604, FSRC = M6-R7, AC = ACC
*****

```

015622 104400
015624 000410

SCOPE
BR

TST123

;BRANCH OVER INPUT DATA

015626 000252 125252 125252
015632 125253

DTA123: 000252,125252,125252,125253

MACY11 27(732) 17-SEP-76 09:41 PAGE 67

TEST OF ADD, ADD, SUB, SUBD MACY11 27(732) 17-SEP-76 09:41 PAGE 67

TEST SECTION

```

015636 000425 052525 052525 DTB123: 000425,052525,052525,052525
015644 052525

015646 170127 045617 TST123: LDFPS #045617 :LOAD FLOATING POINT STATUS
015652 172467 :77750 :LOAD 000252,125252,125252,125252 INTO ACC
015656 173067 :77754 FP123: SUBD DTB123, ACC :SUBTRACT 000425,052525,052525,052525 FROM ACC
015662 170200 STFPS FPS :STORE FLOATING POINT STATUS
015668 022700 045604 CMP #045604,FPS :CHECK FLOATING POINT STATUS
015674 001401 BEQ .+4 :BRANCH IF OK
015678 104000 HLT :FPS NOT EQUAL TO 045604

015674 174067 163102 STD ACC, ANS1 :STORE DIFFERENCE IN ANS1 THRU ANS4
015680 022767 000000 163074 CMP #000000,ANS1 :CHECK ANS1
015706 001401 BEQ .+4 :BRANCH IF OK
015710 104004 HLT+4 :ANS1 NOT EQUAL TO 000000

015712 022767 000000 163064 CMP #000000,ANS2 :CHECK ANS2
015720 001401 BEQ .+4 :BRANCH IF OK
015722 104004 HLT+4 :ANS2 NOT EQUAL TO 000000

015724 022767 000000 163054 CMP #000000,ANS3 :CHECK ANS3
015732 001401 BEQ .+4 :BRANCH IF OK
015734 104004 HLT+4 :ANS3 NOT EQUAL TO 000000

015736 022767 000000 163044 CMP #000000,ANS4 :CHECK ANS4
015744 001401 BEQ .+4 :BRANCH IF OK
015748 104004 HLT+4 :ANS4 NOT EQUAL TO 000000

```

```

*****
TEST 124: TEST SUBC (SUBTRACT DOUBLE PRECISION)
          040052,125252,125252,125252 - 140125,052525,052525,052525 =
          040300,000000,000000,000000
          FPS = 047600, FSRC = MC-AC1, AC = AC2
*****

```

```

015750 104400 SCOPE
015752 000410 BR TST124 :BRANCH OVER INPUT DATA

015754 040052 125252 125252 DTA124: 040052,125252,125252,125252
015762 125252
015764 140125 052525 052525 DTB124: 140125,052525,052525,052525
015772 052525

015774 170127 047617 TST124: LDFPS #047617 :LOAD FLOATING POINT STATUS
015800 172667 :77750 :LOAD 040052,125252,125252,125252 INTO ACC
015804 172567 :77754 :LOAD 140125,052525,052525,052525 INTO ACC
015810 173201 SUBC AC1, AC2 :SUBTRACT AC1 FROM AC2
015812 170200 STFPS FPS :STORE FLOATING POINT STATUS
015814 022700 047600 CMP #047600,FPS :CHECK FLOATING POINT STATUS
015820 001401 BEQ .+4 :BRANCH IF OK
015822 104000 HLT+4 :FPS NOT EQUAL TO 047600

015824 174067 162752 STD ACC, ANS1 :STORE DIFFERENCE IN ANS1 THRU ANS4
015830 022767 040000 162744 CMP #040000,ANS1 :CHECK ANS1
015836 001401 BEQ .+4 :BRANCH IF OK

```

```

016040 104004          HLT+4          :ANS1 NOT EQUAL TO 040300
016042 022767 000000 162734  CMP          #000000,ANS2  :CHECK ANS2
016044 001401          BEQ          .+4      :BRANCH IF OK
016046 104004          HLT+4          :ANS2 NOT EQUAL TO 000000
016054 022767 000000 162724  CMP          #000000,ANS3  :CHECK ANS3
016056 001401          BEQ          .+4      :BRANCH IF OK
016058 104004          HLT+4          :ANS3 NOT EQUAL TO 000000
016066 022767 000000 162714  CMP          #000000,ANS4  :CHECK ANS4
016068 001401          BEQ          .+4      :BRANCH IF OK
016070 104004          HLT+4          :ANS4 NOT EQUAL TO 000000
016100 174167 162706          STD          AC1,ANS5  :STORE AC1 IN ANS5 THRU ANS8
016104 022767 140125 162700  CMP          #140125,ANS5  :CHECK ANS5
016106 001401          BEQ          .+4      :BRANCH IF OK
016108 104010          HLT+10         :AC1 CHANGED
016116 022767 052525 162670  CMP          #052525,ANS6  :CHECK ANS6
016118 001401          BEQ          .+4      :BRANCH IF OK
016120 104010          HLT+10         :AC1 CHANGED
016130 022767 052525 162660  CMP          #052525,ANS7  :CHECK ANS7
016132 001401          BEQ          .+4      :BRANCH IF OK
016134 104010          HLT+10         :AC1 CHANGED
016142 022767 052525 162650  CMP          #052525,ANS8  :CHECK ANS8
016144 001401          BEQ          .+4      :BRANCH IF OK
016146 104010          HLT+10         :AC1 CHANGED

```

```

*****
:TEST 125:      TEST SUBC (SUBTRACT DOUBLE PRECISION)
:              000200,000000,000000,000001 - 000200,000000,000000,000000 =
:              062400,000000,000000,000000
:              FPS = 147600,   FSRC = NO-AC0,   AC = AC3
:              FEC = 12,     FEA = FPI125
*****

```

```

016154 104400          SCOPE
016156 000410          BR          TST125      :BRANCH OVER INPUT DATA
016160 000200 000000 000000 000001 DTB125: 000200,000000,000000,000001
016162 000001
016164 000200 000000 000000 000000 DTB125: 000200,000000,000000,000000
016166 000000
016168 170127 047617 *ST125: LDFPS      #047617      :LOAD FLOATING POINT STATUS
016170 172767 177750 LDD          DTB125, AC3  :LOAD 000200,000000,000000,000000 INTO AC3
016172 172467 177754 LDD          DTB125, AC0  :LOAD 000200,000000,000000,000000 INTO AC0
016174 173200          SUBC         ACC, AC3    :SUBTRACT AC0 FROM AC3
016176 170200          STFPS      FPS       :STORE FLOATING POINT STATUS
016178 170267 162576          STFEC      FEC       :STORE EXCEPTION CODES
016180 022700 147600          CMP          #147600,FPS  :CHECK FLOATING POINT STATUS
016182 001401          BEQ          .+4      :BRANCH IF OK

```

E06

MAINDEC-11-20FPO-0
20FPO.P11

TEST OF ADDF, ADDD, SJBF, SUBD
TEST SECTION

MACY11 27(732) 17-SEP-76 09:41 PAGE 69

016232	104000			HLT			:FMS NOT EQUAL TO 147600
016234	022767	000012	162560	CMP	#12,	FEC	:CHECK FLOATING EXCEPTION CODE
016242	001401			BEG	.+4		:BRANCH IF OK
016244	104000			HLT			:FEC NOT EQUAL TO 12
016246	174367	162530		STD	ACC,	ANS1	:STORE DIFFERENCE IN ANS1 THRU ANS-
016250	022767	062400	162522	CMP	#062400,	ANS1	:CHECK ANS1
016252	001401			BEG	.+4		:BRANCH IF OK
016254	104004			HLT+4			:ANS1 NOT EQUAL TO 062400
016264	022767	000000	162512	CMP	#000000,	ANS2	:CHECK ANS2
016272	001401			BEG	.+4		:BRANCH IF OK
016274	104004			HLT+4			:ANS2 NOT EQUAL TO 000000
016276	022767	000000	162502	CMP	#000000,	ANS3	:CHECK ANS3
016304	001401			BEG	.+4		:BRANCH IF OK
016306	104004			HLT+4			:ANS3 NOT EQUAL TO 000000
016310	022767	000000	162472	CMP	#000000,	ANS4	:CHECK ANS4
016316	001401			BEG	.+4		:BRANCH IF OK
016320	104004			HLT+4			:ANS4 NOT EQUAL TO 000000
016322	174367	162464		STD	ACC,	ANS5	:STORE ACC IN ANS5 THRU ANS6
016326	022767	000200	162456	CMP	#000200,	ANS5	:CHECK ANS5
016334	001401			BEG	.+4		:BRANCH IF OK
016336	104010			HLT+10			:ACC CHANGED
016340	022767	000000	162446	CMP	#000000,	ANS6	:CHECK ANS6
016346	001401			BEG	.+4		:BRANCH IF OK
016350	104010			HLT+10			:ACC CHANGED
016352	022767	000000	162436	CMP	#000000,	ANS7	:CHECK ANS7
016360	001401			BEG	.+4		:BRANCH IF OK
016362	104010			HLT+10			:ACC CHANGED
016364	022767	000000	162426	CMP	#000000,	ANS8	:CHECK ANS8
016372	001401			BEG	.+4		:BRANCH IF OK
016374	104010			HLT+10			:ACC CHANGED

016376	104400			DONE:	SCOPE		
016400	032737	002000	177570		BIT	#SW10,2#SWR	:RING THE BELL
016406	001005				BNE	IS	:NO!
016410	012767	000207	001242		MOV	#BELL,TYPE	:TYPE A BELL
016416	000004	017660			TYPE	TYPE	
016422	005046			18:	CLR	-(6)	:CLEAR TRACE TRAP
016424	032737	010000	177570		BIT	#SW12,2#SWR	:RUN WITH TRT?
016428	001010				BNE	25	
016434	005167	001222			COM	TRPB	
016440	100005				BPL	25	
016442	052716	000020			BIS	#20,(6)	:SET TRACE TRAP
016446	012746	001062			MOV	#BEGIN,-(6)	:JUMP TO START OF TEST
016452	000412				BR	YESRT	
016454	012746	001062		28:	MOV	#BEGIN,-(6)	:JUMP TO START OF TEST
016460	013700	000042			MOV	#42,PC	:GET MONITOR ADDRESS
016464	001404				BEQ	25	:IF NONE
016466	004710				JSR	MON	:GO TO MONITOR
016470	000240				NOP		
016472	000240				NOP		
016474	000240				NOP		
016476	000002			38:	RTI		
016500	000002			YESRT:	RTI		:RETURN TO PROGRAM FROM TRAP
016502	032737	000400	177570	.EMT:	BIT	#SW08,2#SWR	:KILL LDUB OR LOOP ON SPEC. TEST
016510	001404				BEQ	IS	
016512	123767	177570	162260		CMPB	2#SWR,ICNT	:ON RIGHT TEST? *SW7-C*
016520	001437				BEQ	OVER	
016522	113703	177570		18:	MOV8	2#SWR,R3	:GET LB BITS
016526	170003				LDUB		
016530	032737	040000	177570		BIT	#SW14,2#SWR	:LOOP ON TEST
016536	001026				BNE	KIT	
016540	032737	004000	177570		BIT	#SW11,2#SWR	:KILL ITERATIONS
016546	001012				BNE	SAVLAD	
016550	105767	162225			TSTB	ICNT+1	
016554	001404				BEQ	25	:BRANCH IF FIRST
016556	126767	001106	162215		CMPB	TIMES,ICNT+1	:DONE?
016564	001013				BNE	KIT	:BRANCH IF NOT
016566	112767	000001	162205	28:	MOV8	#1,ICNT+1	:FIRST ITERATION
016574	105267	162200		SAVLAD:	INCB	ICNT	:COUNT TEST NUMBERS
016600	011667	001060			MOV	(6),LAD	:SAVE LOOP ADDRESS
016604	016737	162170	177570		MOV	ICNT,2#DISPLAY	:DISPLAY TEST NO. AND ITERATION COUNT
016612	000002				RTI		:RETURN
016614	105267	162161		KIT:	INCB	ICNT+1	
016620	016737	162154	177570	OPER:	MOV	ICNT,2#DISPLAY	:SET UP DISPLAY
016626	005767	001032			LAD	LAD	:FIRST ONE
016632	001760				BEQ	SAVLAD	
016634	016710	001024			MOV	LAD,(6)	:FUDGE RETURN ADDRESS
016640	000002				RTI		:FINISH

016642	032737	002000	177570	.TRP:	BIT	#SW10,2#SWR	:BELL ON ERROR?
016650	001405				BEQ	IS	:NO - SKIP
016652	012767	000207	001000		MOV	#BELL,TYPE	:TYPE A BELL
016660	000004	017660			TYPE	TYPE	
016664	004767	000406		13:	JSR	PC,ERROR	:COUNT THE NUMBER OF ERRORS
016670	010446				MOV	R4, -(6)	
016672	032737	020000	177570		BIT	#SW13,2#SWR	:SKIP TYPEOUT IF SET
016674	001072				BNE	45	
016676	000004	017626			TYPE	RETURN	
016678	016646	000002			MOV	2(6) -(6)	:PUT ADDRESS OF INSTRUCTION ON STACK
016680	162716	000032			SUB	#2,(6)	
016682	011605				MOV	(6) TTY	:TYPE (6) IN OCTAL
016684	004767	000212			JSR	%7,PRINTR	:TYPE LEADING ZERO'S
016686	000004	017634			TYPE	SPACE+3	
016688	010005				MOV	R0, TTY	:TYPE R0 IN OCTAL
016690	004767	000200			JSR	%7,PRINTR	:TYPE LEADING ZERO'S
016692	000004	017635			TYPE	SPACE+4	
016694	012703	001332			MOV	#ANS1,R3	:ADDRESS OF DATA
016696	113604				MOVB	2(6)+,R4	:AMOUNT OF DATA IN TABLE
016698	001426				BEQ	35	
016700	100016				BP,	25	:TYPE STACK?
016702	016667	000006	162020		MOV	6(6),ANS1	
016704	016667	000010	162014		MOV	10(6),ANS2	
016706	016667	000012	162010		MOV	12(6),ANS3	
016708	016667	000014	162004		MOV	14(6),ANS4	
016710	042704	177600			BIC	#177600,R4	:CLEAR SIGN
016712	000004	017635		28:	TYPE	SPACE+4	
016714	012305				MOV	(3)+,TTY	:TYPE (3) + IN OCTAL
016716	004767	000114			JSR	%7,PRINTR	:TYPE LEADING ZERO'S
016718	005304				R4	R4	
016720	001371				ST	R5	
016722	005700			38:	ST	R5	
016724	100016				BP,	45	
016726	000004	017631			ST	SPACE	
016728	170367	161750			ST	FEC	
016730	016705	161754			MOV	FEC,TTY	:TYPE FEC IN OCTAL
016732	004767	000064			JSR	%7,PRINTR	:TYPE LEADING ZERO'S
016734	000004	017634			TYPE	SPACE+3	
016736	016705	161742			MOV	FEA,TTY	:TYPE FEA IN OCTAL
016738	004767	000050			JSR	%7,PRINTR	:TYPE LEADING ZERO'S
016740	012604			48:	MOV	(6)+,R4	
016742	005737	177570			ST	2#SWR	:HALT ON ERROR
016744	100001				BP,	+.4	:SKIP IF CONTINUE
016746	000000				HALT		:HALT ON ERROR!
016748	032737	001000	177570		BITY	#SW09,2#SWR	:CHECK FOR INHIBIT LOOP ON ERROR
016750	001001				BNE	+.4	:SKIP IF LOOP ON ERROR
016752	000002				BITY		
016754	105067	161663			CLRB	ICNT+1	
016756	032737	003400	177570		BITY	#SW08,2#SWR	:CHECK FOR LOAD MICROBREAK
016758	001001				BNE	K,TTY	:BRANCH IF NOT
016760	003733	177570			MOVB	2#SWR,R3	:PUT MICROBREAK ADDRESS IN R3
016762	000000				CLRB		:LOAD MICROBREAK
016764	000000				BITY	K,TTY	:LOOP ON TEST UNTIL NO ERRORS


```

0172136 112767 000001 000130 PRINTR: MOVB #1,R4S :SET ZERO FILL SWITCH
0172144 000402 BR .+6
0172146 005067 000122 PRINTS: CLR R4S :SUPPRESS LEADING ZERO'S
0172152 112767 177772 000115 MOVB #5,R4S+1 :SET COUNT
0172160 010446 MOV R4,-(6) :SAVE R4
0172162 012704 C17264 MOV #3,R4 :SET POINTER TO FIRST ASCII CHAR.
0172166 105014 CLRB (4) :CLEAR FIRST BYTE
0172170 000405 BR 25 :ROTATE FIRST BIT
0172174 105014 :S: CLRB (4) :CLEAR BYTE OF CHARACTER
0172178 006105 PCL TTY :ROTATE BIT INTO C
0172182 106114 ROLB (4) :PACK IT
0172186 006105 ROL (4) :ROTATE BIT INTO C
0172190 106114 25: ROL (4) :PACK IT
0172194 006105 ROLB (4) :ROTATE BIT INTO C
0172198 106114 ROLB (4) :PACK IT
0172202 105714 STB :
0172206 001402 BEQ .+6
0172210 105267 000054 INCB R4S+6
0172214 105767 00005C TSTB R4S+6 :CHECK FILL SWITCH
0172220 001402 BEQ .+6
0172224 152724 00006C BITB R4S+6 :MAKE INTO ASCII CHAR
0172228 105267 000037 INCB R4S+6 :REPEAT
0172232 001355 BNE .+6
0172236 022704 C17264 CMP #0,R4
0172240 001002 BNE .+6
0172244 112724 00006C MOVB #0,R4+
0172248 105014 CLARB (4)
0172252 000004 C17264 TYPE R4S :TYPE IT
0172256 012604 MOV #0,R4 :RES+ORE R4
0172260 000207 RTS
0172264 000004 35: .BLKW 4
0172274 000000 R4S: 0

0172276 005267 000364 ERROR: INC ERRORS :COUNT ERRORS
017302 132737 000001 000041 BITB #1,R4S+1 :AU+C MODE?
017310 001412 BEQ .+1 :NO!
017312 022767 000010 000346 CMP #10,ERRORS :TOO MANY?
017320 001006 BNE .+6 :NOT YET
017322 013700 000042 MOV #42,R0 :GET ADDRESS
017326 001403 BEQ .+3 :FORGET IT IF ZERO
017330 005037 000042 CLRB R4S :CLEAR
017334 004710 :S: CALL THE MONITOR
017338 000207 RTS :END

```

MACY11 27(732) 17-SEP-76 09:41 PAGE 73

TEST OF ADD, ADD, SUB, SUBD
POWER DOWN AND UP ROUTINES

MACY11 27(732) 17-SEP-76 09:41 PAGE 73

```

017340 012777 017534 000306 POWDOWN: MOV #ILLUP,DOWNVEC :SET FOR FAST UP
017346 012777 000340 000302 MOV #340,DOWNVEC+2 :PRIO:7
017354 170246 STFPS -(6) :GET THE FPS
017356 170011 SETD :
017360 174046 STD AC0, -(6) :SAVE AC'S
017362 174146 STD AC1, -(6)
017364 174246 STD AC2, -(6)
017366 174346 STD AC3, -(6)
017370 172404 LDD AC4,AC0
017372 174046 STD AC0, -(6)
017374 172405 LDD AC5,AC0
017376 174046 STD AC0, -(6)
017400 010046 MOV R0, -(6) :SAVE REGISTERS
017402 010146 MOV R1, -(6)
017404 010246 MOV R2, -(6)
017406 010346 MOV R3, -(6)
017410 010446 MOV R4, -(6)
017412 010546 MOV R5, -(6)
017414 010667 000220 000226 MOV SP,SAVE6 :SAVE SP
017420 012777 017430 000226 MOV #POWUP,DOWNVEC :SET UP VECTOR
017426 000000 HALT

017430 016706 000204 POWUP: MOV SAVE6,SP :GET SP
017434 005001 CLR R1 :WAIT LOOP FOR THE TTY
017436 005201 IS: INC R1
017440 001376 BNE IS
017442 012605 MOV (6)+,R5 :GET THE REGISTERS
017444 012604 MOV (6)+,R4
017446 012603 MOV (6)+,R3
017450 012602 MOV (6)+,R2
017452 012601 MOV (6)+,R1
017454 012600 MOV (6)+,R0
017456 170011 SETD
017460 172426 LDD (6)+,AC0 :RESTORE THE AC'S
017462 174005 STD AC0,AC5
017464 172426 LDD (6)+,AC0
017466 174004 STD AC0,AC4
017470 172725 LDD (6)+,AC3
017472 172626 LDD (6)+,AC2
017474 172526 LDD (6)+,AC1
017476 172426 LDD (6)+,AC0
017500 170126 LDFPS (6)+ :RESTORE FPS
017502 012777 017340 000140 MOV #POWDOWN,DOWNVEC :SET UP THE POWER DOWN VECTOR
017510 012777 000340 000134 MOV #340,DOWNVEC+2
017516 000004 017522 TYPE :ASCIZ <15><12>"POWER"
017532 000002 RTI

017534 000000 ILLUP: HALT :THE POWER UP SEQUENCE WAS STARTED
017536 000776 BR :BEFORE THE POWER DOWN WAS COMPLETE

```

```

017540 010546          .IOT:  MOV   TTY, -(6)      ;SAVE TTY
017542 017605 000002          MOV   @2(6), TTY      ;GET ADDRESS TO BE TYPED
017546 105715          1$:   TSTB  (TTY)          ;TERMINATOR?
017550 001406          BEQ   2$              ;
017552 112537 177566          MOVB  (TTY)+, @#177566 ;LOAD AND TYPE THE CHARACTER
017556 105737 177564          TSTB  @#177564      ;IS THE PRINTER READY
017562 100375          BPL   .-4            ;
017564 000770          BR    1$             ;GET THE NEXT CHARACTER
017566 017646 000002          2$:  MOV   @2(6), -(6) ;GET ADDRESS TO BE TYPED
017572 062766 000002 000004  ADD   #2, 4(6)        ;ADD 2 TO THE ADDRESS
017600 022666 000002          CMP   (6)+, 2(6)    ;IS IT .+2?
017604 001006          BNE   3$             ;NO
017606 062705 000002          ADD   #2, TTY        ;ADD 2 TO THE ADDRESS
017612 042705 000001          BIC   #1, TTY        ;BACK UP TO AN EVEN BYTE
017616 010566 000002          MOV   TTY, 2(6)     ;RESTORE ADDRESS
017622 012605          3$:  MOV   (6)+, TTY     ;RESTORE TTY
017624 000002          RTI                  ;RETURN

017626 005015          000          RETURN: .ASCIZ  <15><12>  ;RETURN AND LINEFEED
017631 015          020012 020040  SPACE: .ASCIZ  <15><12>  ;RETURN AND 3 SPACES
017636 000

017640 017640          .EVEN
017642 000000          SAVE6: 0
017644 172160          FPTADR: 172160      ;FLOATING POINT ADDRESS ON THE 11 20
017646 000244 000246          FPVECT: 244, 246   ;FLOATING POINT VECTOR ADDRESS
017650 000024 000026          DWNVEC: 24, 26     ;POWER DOWN VECTOR ADDRESS
017654 000024 000026          UPVEC:  24, 26    ;POWER UP VECTOR ADDRESS
017660 000000          .TYPE: 0
017662 000000          TRPB:  0
017664 000000          LAD:   0            ;LOOP ADDRESS
017666 000000          ERRORS: 0          ;ERROR COUNT
017670 000377          TIMES: 377        ;ITERATION COUNT
000000:          .END

```

K06

AC1	AC2	AC3	AC4	AC5	AC6	AC7	AC8	AC9	AC10	AC11	AC12	AC13	AC14
AC1 =000000	AC2 =000001	AC3 =000002	AC4 =000003	AC5 =000004	AC6 =000005	AC7 001002	AC8 001004	AC9 001006	AC10 001010				
391*	392*	393*	394*	395*	396*	425*	426*	427*	428*				
1126*	984*	952*	862*	1541*	2005*	2853*	3510*	395*	396*	425*	660*	840*	1061*
2401*	1806*	1719*	1719*	2370*	2763*	3390*	394*	862*	1541*	2005*	2853*	3510*	395*
2401*	2257*	2761*	3560*	3930*	392*	534*	535*	541	623*	624*	630	742*	743*
2407	3055*	3056*	3062	3137*	3138*	3144	3899	3903*	3904	3905*	3906	3928*	3929
2501*	2099*	2101*	2101*	2101*	2101*	2101*	2101*	2101*	2101*	2101*	2101*	2101*	2101*
2502*	2310*	2311*	2317	2430*	2431*	2437	2581*	2582*	2597	2660*	2661*	2667	2667
2502*	2774	3014*	3015*	3021	3096*	3097*	3103	3178*	3179*	3185	3185	3559*	3559*
2507*	3566	3642*	3643	3666	3900	3934*	393*	504*	505*	511	594*	595*	601
2507*	952*	960	1125*	1127*	1138	1173*	1174*	1180	1296*	1297*	1303	1337*	1337*
2507*	1338*	1344	1417*	1418*	1424	1458*	1459*	1465	1499*	1500*	1506	1703*	1704*
2507*	1719	1754*	1755*	1770	1907*	1908*	1923	2039*	2041*	2047	2100*	2101	2129
2507*	2370*	2371*	2377	2461*	2462*	2477	2590*	2691*	2697	2721*	2722	2737	2762*
2507*	2763	2783	2809*	2810*	2816	2891*	2892*	2898	2973*	2974*	2980	3393*	3394*
2507*	3390	3424*	3425*	3431	3641*	3643*	3649	3901	3933*	3933*	3933*	3933*	3933*
2507*	394*	474*	475*	481	564*	565*	571	653*	654*	660	713*	714*	720
2507*	862*	863*	869	892*	893*	899	1085*	1086	1101	1255*	1256*	1262	1540*
2507*	1541*	1547	1581*	1582*	1588	1856*	1857*	1872	1957*	1958*	1964	1998*	1999*
2507*	2005*	2040*	2041	2064	2220*	2221*	2227	2280*	2281*	2287	2720*	2722*	2729
2507*	2853*	2851*	2857	3219*	3220*	3226	3301*	3302*	3308	3465*	3467*	3473	3508*
2507*	3510*	3525	3701*	3703*	3714	3902	3932*	3932*	3932*	3932*	3932*	3932*	3932*
2507*	395*	3903	3931*	3929*	511*	512	541*	542	571*	572	601*	602	630*
2507*	425*	481*	482	511*	512	541*	542	571*	572	601*	602	630*	631
2507*	660*	661	690*	691	720*	721	749*	750	779*	780	809*	810	839*
2507*	840	869*	870	899*	900	929*	930	960*	961	1001*	1002	1031*	1032
2507*	1061*	1062	1092*	1093	1138*	1139	1180*	1181	1221*	1222	1262*	1263	1303*
2507*	1304	1344*	1345	1385*	1386	1424*	1425	1465*	1466	1506*	1507	1547*	1548
2507*	1588*	1589	1627*	1628	1668*	1669	1719*	1720	1770*	1771	1821*	1822	1872*
2507*	1873	1923*	1924	1964*	1965	2005*	2006	2047*	2048	2112*	2113	2167*	2168
2507*	2197*	2198	2227*	2228	2257*	2258	2287*	2288	2317*	2318	2347*	2348	2377*
2507*	2378	2407*	2408	2437*	2438	2477*	2478	2517*	2518	2557*	2558	2597*	2598
2507*	2637*	2638	2667*	2668	2697*	2698	2728*	2729	2774*	2775	2816*	2817	2857*
2507*	2858	2898*	2899	2939*	2940	2990*	2991	3021*	3022	3062*	3063	3103*	3104
2507*	3144*	3145	3185*	3186	3226*	3227	3267*	3268	3308*	3309	3349*	3350	3390*
2507*	3391	3431*	3432	3473*	3474	3525*	3526	3566*	3567	3607*	3608	3649*	3650
2507*	3714*	3715	3815	3819*	516	606	635	665	695	725	754	794	814
2507*	426*	486	516	546	606	635	665	695	725	754	794	814	834
2507*	844	874	904	934	1006	1036	1066	1097	1143	1185	1226	1267	1307
2507*	1308	1349	1390	1429	1470	1511	1552	1593	1632	1673	1724	1775	1826
2507*	1877	1928	1969	2010	2052	2117	2172	2202	2232	2262	2292	2322	2352
2507*	2382	2412	2442	2482	2522	2562	2602	2642	2672	2702	2732	2779	2821
2507*	2862	2903	2944	2985	3026	3067	3108	3149	3190	3231	3272	3313	3354
2507*	3395	3436	3478	3530	3571	3612	3654	3719	3780*	3820*	3860*	3900*	3940*
2507*	427*	1101*	1102	1147*	1148	1189	1230	1271	1312	1353	1394	1433	1474
2507*	1515	1556	1597	1636	1677	1728	1779	1830	1881	1932	1973	2014	2056
2507*	2121	2737*	2738	2783*	2784	2825	2866	2907	2948	2989	3030	3071	3112
2507*	3153	3194	3235	3276	3317	3358	3399	3440	3482	3534	3575	3616	3659
2507*	3723	3821*	1152	1193	1234	1275	1316	1357	1398	1437	1478	1519	1560
2507*	428*	1136	1152	1193	1234	1275	1316	1357	1398	1437	1478	1519	1560
2507*	1601	1640	1681	1732	1783	1834	1885	1936	1977	2018	2060	2101	2142
2507*	2786	2829	2870	2911	2952	2993	3034	3075	3116	3157	3198	3239	3280

DTA34	004604	1372#	1378
DTA35	004732	1413#	1417
DTA36	005050	1452#	1458
DTA37	005176	1493#	1499
DTA4	001456	560#	564
DTA40	005324	1534#	1540
DTA41	005452	1575#	1581
DTA42	005600	1616#	1620
DTA43	005716	1655#	1661
DTA44	006044	1697#	1703
DTA45	006222	1748#	1754
DTA46	006400	1799#	1805
DTA47	006556	1850#	1856
DTA5	001550	590#	594
DTA50	006734	1901#	1907
DTA51	007112	1951#	1957
DTA52	007240	1992#	1998
DTA53	007366	2033#	2039
DTA54	007572	2093#	2099
DTA55	010014	2156#	2160
DTA56	010106	2186#	2190
DTA57	010200	2216#	2220
DTA6	001642	620#	623
DTA60	010272	2246#	2250
DTA61	010364	2276#	2280
DTA62	010456	2306#	2310
DTA63	010550	2336#	2340
DTA64	010642	2366#	2370
DTA65	010734	2396#	2400
DTA66	011026	2426#	2430
DTA67	011120	2457#	2461
DTA7	001730	649#	653
DTA70	011242	2497#	2501
DTA71	011364	2537#	2541
DTA72	011506	2577#	2581
DTA73	011630	2617#	2621
DTA74	011752	2656#	2660
DTA75	012044	2686#	2690
DTA76	012136	2716#	2720
DTA77	012262	2757#	2761
DTB1	001204	471#	475
DTB10	002026	680#	684
DTB100	012434	2805#	2810
DTB101	012562	2846#	2851
DTB102	012710	2887#	2892
DTB103	013036	2928#	2933
DTB104	013164	2969#	2974
DTB105	013312	3010#	3015
DTB106	013440	3051#	3056
DTB107	013566	3092#	3097
DTB11	002120	710#	714
DTB110	013714	3133#	3138
DTB111	014042	3174#	3179
DTB112	014170	3215#	3220
DTB113	014316	3256#	3261
DTB114	014444	3297#	3302

DTB115	014572	3338#	3343
DTB116	014720	3379#	3384
DTB117	015046	3420#	3425
DTB120	015174	3461#	3467
DTB121	015326	3504#	3510
DTB122	015510	3555#	3560
DTB123	015636	3596#	3601
DTB124	015764	3637#	3642
DTB125	016170	3697#	3702
DTB13	002300	769#	773
DTB14	002372	799#	803
DTB15	002464	829#	833
DTB16	002556	859#	863
DTB17	002650	889#	893
DTB2	001276	501#	505
DTB20	002742	919#	923
DTB21	003034	949#	954
DTB22	003132	981#	986
DTB23	003260	1021#	1025
DTB24	003352	1051#	1055
DTB25	003444	1081#	1085
DTB26	003570	1122#	1126
DTB27	003736	1169#	1174
DTB3	001370	531#	535
DTB30	004064	1210#	1215
DTB31	004212	1251#	1256
DTB32	004340	1292#	1297
DTB33	004466	1333#	1338
DTB34	004614	1374#	1379
DTB36	005060	1454#	1459
DTB37	005206	1495#	1500
DTB4	001462	561#	565
DTB40	005334	1536#	1541
DTB41	005462	1577#	1582
DTB43	005726	1657#	1662
DTB44	006054	1699#	1704
DTB45	006232	1750#	1755
DTB46	006410	1801#	1806
DTB47	006566	1852#	1857
DTB5	001554	591#	595
DTB50	006744	1903#	1908
DTB51	007122	1953#	1958
DTB52	007250	1994#	1999
DTB53	007376	2035#	2040
DTB54	007602	2095#	2100
DTB55	010020	2157#	2161
DTB56	010112	2187#	2191
DTB57	010204	2217#	2221
DTB60	010276	2247#	2251
DTB61	010370	2277#	2281
DTB62	010462	2307#	2311
DTB63	010554	2337#	2341
DTB64	010646	2367#	2371
DTB65	010740	2397#	2401
DTB66	011032	2427#	2431
DTB67	011124	2457#	2461

CROSS REFERENCE TABLE		TEST OF	ADDF	ADDD	SUBF	SUBD	MACRO NAMES											
3812	3825	3833	3836				555	555	685	715	744	774	805	835	865	895	925	955
1228	1279	1320	1361	1441	1482	1523	1564	1644	1685	1736	1787	1838	1889	1940	1991	2042	2093	2144
550	580	639	669	699	758	788	848	878	929	970	1021	1072	1123	1174	1225	1276	1327	1378
2974	2915	2956	2997	3038	3079	3120	3161	3222	3273	3324	3375	3426	3477	3528	3579	3630	3681	3732
2206	2236	2266	2296	2326	2356	2386	2416	2446	2476	2506	2536	2566	2596	2626	2656	2686	2716	2746

MAYNDEC-11-22CPD-C CROSS REFERENCE TABLE -- PERMANENT SYMBOLS MACY11 27(732) 17-SEP-76 C9:41 PAGE 89

PERM SYMBOL	ADD. SYMBOL	ADD. SYMBOL	ADD. SYMBOL	ADD. SYMBOL	ADD. SYMBOL	ADD. SYMBOL	ADD. SYMBOL	ADD. SYMBOL	ADD. SYMBOL	ADD. SYMBOL	ADD. SYMBOL	ADD. SYMBOL	ADD. SYMBOL	ADD. SYMBOL																								
1459	1500	1541	1582	1621	1662	1704	1755	3772	3778	3780	3785	3798	3804	3842	3846	3872	3975	3989	3992	3998	3824	3828	3843	3947	3872	3975	3989	3992	3998	3824	3828	3843	3947	3872	3975	3989	3992	3998

1433	1433	1433	1461	1466	1470	1474	1478	1502	1507	1511	1515	1519	1543	1543
1556	1556	1560	1584	1589	1593	1597	1601	1623	1628	1632	1636	1640	1664	1664
1783	1783	1809	1707	1711	1715	1720	1724	1728	1732	1758	1762	1766	1771	1771
1911	1911	1915	1813	1817	1822	1826	1830	1834	1860	1864	1868	1872	1877	1877
2014	2014	2018	1919	1924	1928	1932	1936	1960	1965	1969	1973	1977	2001	2001
2111	2111	2115	2043	2048	2052	2056	2060	2055	2069	2073	2077	2104	2108	2108
2211	2211	2215	2130	2134	2138	2142	2163	2168	2172	2192	2198	2204	2208	2208
2311	2311	2315	2262	2267	2271	2275	2313	2318	2322	2342	2348	2354	2358	2358
2411	2411	2415	2408	2413	2417	2421	2465	2469	2473	2478	2482	2486	2490	2490
2511	2511	2515	2549	2553	2557	2561	2585	2589	2593	2598	2602	2606	2610	2610
2611	2611	2615	2663	2667	2671	2675	2702	2706	2710	2714	2718	2722	2726	2726
2711	2711	2715	2784	2788	2792	2796	2825	2829	2833	2837	2841	2845	2849	2849
2811	2811	2815	2907	2911	2915	2919	2948	2952	2956	2960	2964	2968	2972	2972
2911	2911	2915	3030	3034	3038	3042	3071	3075	3079	3083	3087	3091	3095	3095
3011	3011	3015	3157	3161	3165	3169	3194	3198	3202	3206	3210	3214	3218	3218
3111	3111	3115	3290	3294	3298	3302	3317	3321	3325	3329	3333	3337	3341	3341
3211	3211	3215	3403	3407	3411	3415	3440	3444	3448	3452	3456	3460	3464	3464
3311	3311	3315	3530	3534	3538	3542	3567	3571	3575	3579	3583	3587	3591	3591
3411	3411	3415	3654	3658	3662	3666	3691	3695	3699	3703	3707	3711	3715	3715
3511	3511	3515	3740	3744	3748	3752	3777	3781	3785	3789	3793	3797	3801	3801
3611	3611	3615	3841	3845	3849	3853	3878	3882	3886	3890	3894	3898	3902	3902
3711	3711	3715	3868	3872	3876	3880	3905	3909	3913	3917	3921	3925	3929	3929
3811	3811	3815	3915	3919	3923	3927	3952	3956	3960	3964	3968	3972	3976	3976
3911	3911	3915	3942	3946	3950	3954	3979	3983	3987	3991	3995	3999	4003	4003
4011	4011	4015	3942	3946	3950	3954	3979	3983	3987	3991	3995	3999	4003	4003
4111	4111	4115	3942	3946	3950	3954	3979	3983	3987	3991	3995	3999	4003	4003
4211	4211	4215	3942	3946	3950	3954	3979	3983	3987	3991	3995	3999	4003	4003
4311	4311	4315	3942	3946	3950	3954	3979	3983	3987	3991	3995	3999	4003	4003
4411	4411	4415	3942	3946	3950	3954	3979	3983	3987	3991	3995	3999	4003	4003
4511	4511	4515	3942	3946	3950	3954	3979	3983	3987	3991	3995	3999	4003	4003
4611	4611	4615	3942	3946	3950	3954	3979	3983	3987	3991	3995	3999	4003	4003
4711	4711	4715	3942	3946	3950	3954	3979	3983	3987	3991	3995	3999	4003	4003
4811	4811	4815	3942	3946	3950	3954	3979	3983	3987	3991	3995	3999	4003	4003
4911	4911	4915	3942	3946	3950	3954	3979	3983	3987	3991	3995	3999	4003	4003
5011	5011	5015	3942	3946	3950	3954	3979	3983	3987	3991	3995	3999	4003	4003

TEST OF CROSS REFERENCE	ADD. PERMANENT SYMBOLS	ADD. SUB. SYMBOLS	SUB. SYMBOLS	MACY11 27(732)	17-SEP-76	09:41	PAGE 91
1190	1221	1262	1303	1344	1385	1424	1465
1821	1872	1923	1964	2005	2047	2064	2112
3062	3103	3144	3185	3226	3267	3308	3349
3666	3714	3731	3899	3900	3901	3902	3904
491	511	541	571	601	630	660	690
929	960	1001	1031	1061	1092	1101	1138
2347	2377	2407	2437	2477	2517	2557	2597
416	476	506	536	566	596	625	655
894	924	955	987	1026	1056	1087	1128
1461	1501	1542	1583	1622	1663	1705	1756
2192	2222	2252	2282	2312	2342	2372	2402
2692	2723	2764	2811	2852	2893	2934	2975
3262	3303	3344	3385	3426	3468	3511	3561
419	488	519	579	639	708	779	848
3512	3705	3832	1706	1757	1808	1859	1910
3808							
2810	2851	2892	2933	2974	3015	3056	3097
3425	3467	3510	3560	3601	3643	3703	3763
2151	2191	2231	2251	2281	2311	2341	2371
2661	2691	2722	2763				
377	3794	3829	3839				
428	486	529	546	3949			
594	622	663					
765	777	793	793				
913	925	929	943	956	960	975	989
1075	1088	1092	1116	1130	1138	1162	1176
1285	1299	1303	1326	1340	1344	1367	1381
1488	1502	1506	1529	1543	1547	1570	1584
1692	1707	1719	1743	1758	1770	1794	1809
1946	1960	1964	1987	2001	2005	2028	2043
2181	2193	2197	2211	2223	2227	2241	2253
2333	2343	2347	2361	2373	2377	2391	2403
2492	2505	2517	2532	2545	2557	2572	2585
2681	2693	2697	2711	2724	2728	2752	2766
2880	2894	2898	2921	2935	2939	2962	2976
3085	3099	3103	3126	3140	3144	3167	3181
3290	3304	3308	3331	3345	3349	3372	3386
3497	3513	3525	3548	3562	3566	3589	3603
3940	465	477	495	507	525	537	555
686	704	716	734	745	763	775	793
913	925	943	956	974	988	1015	1027
1176	1203	1217	1244	1258	1285	1299	1326
1488	1502	1529	1543	1570	1584	1611	1623
1808	1844	1859	1895	1910	1946	1960	1987
2181	2193	2211	2223	2241	2253	2271	2283
2403	2421	2433	2451	2464	2491	2504	2531
2681	2693	2711	2724	2751	2765	2798	2812
2880	2894	2898	2921	2935	2939	2962	2976
3085	3099	3103	3126	3140	3144	3167	3181
3290	3304	3308	3331	3345	3349	3372	3386
3497	3513	3525	3548	3562	3566	3589	3603
3940	465	477	495	507	525	537	555
686	704	716	734	745	763	775	793
913	925	943	956	974	988	1015	1027
1176	1203	1217	1244	1258	1285	1299	1326
1488	1502	1529	1543	1570	1584	1611	1623
1808	1844	1859	1895	1910	1946	1960	1987
2181	2193	2211	2223	2241	2253	2271	2283
2403	2421	2433	2451	2464	2491	2504	2531
2681	2693	2711	2724	2751	2765	2798	2812
2880	2894	2898	2921	2935	2939	2962	2976
3085	3099	3103	3126	3140	3144	3167	3181
3290	3304	3308	3331	3345	3349	3372	3386
3497	3513	3525	3548	3562	3566	3589	3603
3940	465	477	495	507	525	537	555
686	704	716	734	745	763	775	793
913	925	943	956	974	988	1015	1027
1176	1203	1217	1244	1258	1285	1299	1326
1488	1502	1529	1543	1570	1584	1611	1623
1808	1844	1859	1895	1910	1946	1960	1987
2181	2193	2211	2223	2241	2253	2271	2283
2403	2421	2433	2451	2464	2491	2504	2531
2681	2693	2711	2724	2751	2765	2798	2812
2880	2894	2898	2921	2935	2939	2962	2976
3085	3099	3103	3126	3140	3144	3167	3181
3290	3304	3308	3331	3345	3349	3372	3386
3497	3513	3525	3548	3562	3566	3589	3603

.PAGE	3603	3630	3645	3689	3705														
.PAGE	481	511	541	571	601	630	660	690	720	749	779	809	839	869	899				
.PAGE	929	960	993	997	1031	1061	1092	1134	1138	1180	1221	1262	1303	1344	1385				
.PAGE	1424	1465	1506	1547	1588	1627	1668	1711	1715	1762	1766	1813	1817	1864	1868				
.PAGE	1915	1919	1964	2005	2047	2108	2112	2167	2197	2227	2257	2287	2317	2347	2377				
.PAGE	2420	2437	2469	2473	2509	2513	2549	2553	2599	2593	2629	2633	2667	2697	2728				
.PAGE	2915	2919	2916	2857	2898	2939	2980	3021	3062	3103	3144	3185	3226	3267	3308				
.PAGE	3210	3290	3431	3473	3517	3521	3566	3607	3649	3710	3714								
.PAGE	3900	372	412	422	460	490	520	550	580	610	639	669	699	729	758				
.PAGE	818	818	848	878	908	938	969	1010	1040	1070	1110	1156	1197	1238	1279				
.PAGE	1320	1361	1402	1441	1482	1523	1564	1605	1644	1685	1736	1787	1838	1889	1940				
.PAGE	1981	2022	2081	2146	2176	2206	2236	2266	2296	2326	2356	2386	2416	2446	2486				
.PAGE	2526	2566	2606	2646	2676	2706	2746	2792	2833	2874	2915	2956	2997	3038	3079				
.PAGE	3120	3161	3202	3243	3284	3325	3366	3407	3448	3490	3542	3583	3624	3683	3748				
.PAGE	3798	3851	3895	3940	3944														
.PAGE	460	3748																	
.PAGE	331																		
.PAGE	327	412																	
.PAGE	328	372	422	460	3748	3798	3851	3895	3944										

ERRORS DETECTED: 0
DEFAULT GLOBALS GENERATED: 0

*.DCFPD.SEG/SOL/CRF/PAGNUM=DCFPD
RUN-TIME: 27 40 6 SECONDS
RUN-TIME RATIO: 2063/74=27.7
CORE USED: 12K (23 PAGES)

