

FP11

MULTIPLY

MD-11-DCFPT-E

EP DCFPT E DL A

OCT 1976

COPYRIGHT © 1976

digital

FICHE 1 OF 1

Made in U.S.A.

This microfiche card contains a grid of frames. The first column on the left contains frames with text and diagrams, likely representing program instructions or data structures. The remaining columns contain frames with dense numerical data, possibly representing the results of a multiplication operation. The data is organized in a structured, tabular format across multiple rows and columns.

.REPT 0

IDENTIFICATION

PRODUCT CODE: MAINDEC-11-DCFPT-E-D
 PRODUCT NAME: FP11 MULTIPLY EXERCISER
 DATE CREATED: 1-NOV-72
 MAINTAINER: DIAGNOSTIC GROUP
 AUTHOR: 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.

110113151617181920212223242526272829303132333435363738394041424344454647484950

108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140

MAINDEC-11-DCFPT-D-D
DESCRIPTION

FP11 MULTIPLY EXERCISER

1. ABSTRACT

THIS PROGRAM EXERCISES THE FP11 FLOATING POINT MULTIPLY INSTRUCTIONS (MULF AND MULD) WITH RANDOM NUMBER PATTERNS. THE ANSWERS ARE CHECKED AGAINST RESULTS OBTAINED USING THE CORRESPONDING FORTRAN SOFTWARE ROUTINES.

2. REQUIREMENTS

2.1 EQUIPMENT

PDP11/45 STANDARD COMPUTER WITH FP11 OPTION

2.2 STORAGE

THE ROUTINES USE MEMORY LOCATIONS 0 - 17500. THE MAP AT THE END OF THE LISTINGS SHOWS THE ABSOLUTE LOCATIONS OF THE FORTRAN MATH ROUTINES WHICH WERE ASSEMBLED SEPARATELY AND LINKED TO THE MAIN PROGRAM VIA LNKX11 ON A DECSYSTEM-10.

2.3 PRELIMINARY PROGRAMS

DCFPA THRU DCFPL

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.

MAINDEC-11-DCFPT-D
DCFPT.E.P11

FLOATING POINT MULTIPLY EXERCISER

EO1

MACY11 27(732) 17-SEP-76 10:19 PAGE 4

141
142

3) SET SWITCHES (SEE 5.1.1) ALL DOWN FOR WORST CASE.
4) PRESS START.

143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198

5) THE PROGRAM WILL LOOP AND BELL WILL RING ONCE EVERY PASS.
6) THE DISPLAY ON THE 11/45 WILL SHOW THE ITERATION COUNT IN THE LEFT BYTE AND TEST NUMBER IN THE RIGHT. TO USE, SET THE DATA DISPLAY SWITCH TO THE DISPLAY POSITION.

5. OPERATING PROCEDURE

5.1 OPERATIONAL SWITCH SETTINGS

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 CORE IMAGE TYPE-OUT (16 BIT WORDS)
 0 FLOATING POINT TYPE-OUT (SIGN, EXPONENT, FRACTION)
SW(08) = 1 LOOP ON TEST IN SW(7:0)
 0 LOAD SW(7:0) INTO UB REGISTER

5.2 SUBROUTINE ABSTRACTS

5.2.1 SCOPE

THIS SUBROUTINE CALL IS PLACED BETWEEN EACH SUBTEST IN THE TEST SECTION. IT RECORDS THE STARTING ADDRESS OF EACH SUBTEST AS IT IS BEING ENTERED IN LOCATION "LAD". IF A SCOPE LOOP IS REQUESTED, THE CURRENT SUBTEST WILL BE LOOPED UPON. SW(11) ON A 1 INHIBITS ITERATION OF SUBTESTS. THE CONTENTS OF LAD MAY BE USED TO DETERMINE THE LAST SUBTEST SUCCESSFULLY COMPLETED. LAD IS UPDATED INSIDE EACH SUBTEST AFTER THE FORTRAN ANSWER IS CALCULATED, SO THAT THE ITERATIONS WILL INCLUDE ONLY THE FP11 PORTION OF THE TEST.

5.2.2 HLT

THIS ROUTINE PRINTS OUT AN ERROR MESSAGE (SEE 6.1). TO INHIBIT TYPEOUTS, PUT SW(13) ON A 1.

5.2.3 TRTRAP

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

MAINDEC-11-DCFPT-D
DCFPT.E.P11

FLOATING POINT MULTIPLY EXERCISER

GO1
MACY11 27(732) 17-SEP-76 10:19 PAGE 6

199

REACHED.

200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300

MAINDEC-11-DCFPT-D-0 FP11 MULTIPLY EXERCISER PAGE 5
DESCRIPTION

5.2.4 TRAPCATCHER

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

5.2.5 FLOATING POINT TRAP (TO 244)

SINCE SOME OF THE SUBTESTS HAVE INTERRUPTS ENABLED, THE FLOATING POINT TRAP (FLTERR) CHECKS TO SEE IF FORTRAN ALSO GOT AN ERROR CONDITION. IF FORTRAN DIDN'T INDICATE AN ERROR, OR INTERRUPTS WERE DISABLED AN ERROR HLT OCCURS (SEE 5.2.2). IF AN INTERRUPT WAS ANTICIPATED, BUT DIDN'T OCCUR, THE SUBTEST WILL DETECT THE ERROR.

6. ERRORS

6.1 ERROR PRINTOUT

THE FORMAT IS AS FOLLOWS:

ADDRESS, OPERAND, OPERATOR, OPERAND, EQUALS
FPP: ANSWER, FPS, FEC, FEA
FORTRAN: ANSWER, FPS, FEC, FEA

WHERE:

ADDRESS = ADDRESS OF ERROR HLT
OPERAND = RANDOM FLOATING POINT NUMBER INPUTS
OPERATOR = ARITHMETIC OPERATOR (+ OR -)
EQUALS = (=)
ANSWER = FLOATING POINT ANSWER
FPS = FLOATING POINT STATUS
FEC = FLOATING EXCEPTION CODES (ERROR CODES)
FEA = FLUATING EXCEPTION ADDRESS (ERROR ADDRESS)

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

6.2 ERROR RECOVERY

RESTART AT 200

7. RESTRICTIONS

NONE

255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310

MAINDEC-11-DCFPT-D-0 FP11 MULTIPLY EXERCISER PAGE 6
DESCRIPTION

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

THIS PROGRAM TESTS THE MULTIPLY INSTRUCTIONS ON THE FP11 IN
ROUND AND TRUNCATE MODES AND WITH INTERRUPTS ON AND OFF.
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. 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.

THE FORTRAN MATH ROUTINES WERE TAKEN UNMODIFIED FROM THE
PDP-11 FORTRAN PACKAGE AND ASSEMBLED AS SEPERATE MODULES.
THEY WERE LINKED TO THE MAIN PROGRAMS VIA LNKX11 ON A
DECSYSTEM-10 WHICH PRODUCES A BINARY TAPE IN THE NORMAL
ABSOLUTE FORMAT. THUS, THE PROGRAMS LOAD AND RUN JUST LIKE
ANY OTHER DIAGNOSTIC PROGRAM.

NOTE: SINCE THE FP11 LOAD, STORE, AND COMPARE
INSTRUCTIONS (LDF, LDD, STF, STD, CMPF, AND CMPD) ARE ALSO
USED IN THIS PROGRAM, IT IS POSSIBLE THAT THEY AND NOT THE
MULTIPLY INSTRUCTIONS COULD CAUSE ERRORS.

.ENDR

000000

```
.TITLE MAINDEC-11-DCFPT-D      FLOATING POINT MULTIPLY EXERCISER
.ASECT
.GLOBL SWLR,SWLD,SERRA
;COPYRIGHT 1972, DIGITAL EQUIPMENT CORP., MAYNARD, MASS
;PROGRAM BY KEN CHAPMAN
```

```

SWITCH      USE
-----      -----
      8      0 - LOAD UB REGISTER WITH SW<7:0>
              1 - LOOP ON TEST IN SW<7:0>
      9      0 - SIGN, EXPONENT, MANTISSA
              1 - CORE IMAGE (16 BIT WORDS)
     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:

```

;ADDRESS, OPERAND, OPERATOR, OPERAND, EQUALS
;FPP: ANSWER, FPS, FEC, FEA
;FORTRAN:
```

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

000000		RO=	X0	
000001		R1=	X1	
000002		R2=	X2	
000003		R3=	X3	
000004		R4=	X4	
000005		R5=	X5	
000005		TTY=	X5	
000006		SP=	X6	
000007		PC=	X7	
000000		AC0=	X0	
000001		AC1=	X1	
000002		AC2=	X2	
000003		AC3=	X3	
000004		AC4=	X4	
000005		AC5=	X5	
000400		SW08=	000400	
001000		SW09=	001000	
002000		SW10=	002000	
004000		SW11=	004000	
010000		SW12=	010000	
020000		SW13=	020000	
040000		SW14=	040000	
100000		SW15=	100000	
177570		SWR=	177570	
177776		PS=	177776	
177570		DISPLAY=	SWR	
000000		DUMMY=	HALT	
000240		NOP=	240	
104400		SCOPE=	TRAP	
104000		HLT=	EMT	
000004		TYPE=	IOT	
000207		BELL=	207	
000000		. =	0	;TRAP CATCHER FROM 0 - 776
000046	000046	. =	46	
000052	004150	\$ENDAD	52	
	000052	. =	52	
		40000		
	000200	. =	200	
000200	000167		JMP	BEG
	001000	. =	1000	
001000	000000	ICNT:	0	
001002	000000	LONUM:	DUMMY	
001004	000000		DUMMY	
001006	000000		DUMMY	
001010	000000		DUMMY	
001012	000000	HINUM:	DUMMY	
001014	000000		DUMMY	
001016	000000		DUMMY	

MAINDEC-11-DCFPT-D
DCFPTC.P11

FLOATING POINT MULTIPLY EXERCISER
ANSWER AREA AND SETUP ROUTINE

L01
MACY11 27(732) 17-SEP-76 10:19 PAGE 11

001020 000000

DUMMY

```
001022 000000      ANS1:  DUMMY
001024 000000      DUMMY
001026 000000      DUMMY
001030 000000      DUMMY

001032 000000      ANS2:  DUMMY
001034 000000      DUMMY
001036 000000      DUMMY
001040 000000      DUMMY

001042 000000      FPS:    0           ;FLOATING POINT STATUS
001044 000000      FEC:    0           ;FLOATING EXCEPTION CODES
001046 000000      FPC:    0           ;FLOATING PC
001050 000000      SFPS:   0           ;FORTRAN FLOATING POINT STATUS
001052 000000      SFEC:   0           ;FORTRAN FLOATING EXCEPTION CODES
001054 000000      SFPC:   0           ;FORTRAN FLOATING PC

001056 012706 000600      BEG:  MOV    #600,SP           ;** STACK AT 600 **
001062 012737 001104 000004      MOV    #M1120,2#4           ;FIND OUT WHICH MACHINE THIS IS
001070 005737 177772      TST   2#177772           ;IS PIRQ THERE?
001074 012767 000006 003062      MOV    #6,YESRT           ;FUDGE IN RTT IF 11/45
001102 000403      BR    BEGIN

001104 016737 005712 000010      M1120: MOV   FPTADR,2#10      ;LOAD THE ILLEGAL INSTRUCTION VECTOR
                                           ;WITH THE ADDRESS OF THE FPU.
                                           ;THE FPU WILL HANDLE THE BAD OPCODES

001112 012737 000006 000004      BEGIN: MOV   #6,2#4           ;RESET 4
001120 012706 000600      MOV   #600,SP
001124 012737 004164 000014      MOV   #YESRT,2#14         ;SET TRACE T-AP VECTOR
001132 012777 006450 005670      MOV   #POWDOWN,2#INVEC
001140 012777 003340 005664      MOV   #340,2#INVEC+2
001146 012737 006650 000020      MOV   #.IOT,2#20         ;SET UP VECTOR 20
001154 012700 000030      MOV   #30,R0              ;SET R0 TO VECTOR 30
001160 012720 005404      MOV   #.TRAP,(0)+         ;SET EMT VECTOR
001164 012720 000340      MOV   #340,(0)+
001170 012720 004166      MOV   #.EMT,(0)+         ;SET TRAP VECTOR
001174 012710 000340      MOV   #340,(0)
001200 012777 005052 005616      MOV   #FLTERR,2#PVECT     ;LOAD INTERRUPT VECTOR
001206 012777 000340 005612      MOV   #340,2#PVECT+2     ;LOCK UP PROCESSOR
001214 005367 177560      CLR   ICNT
001220 005067 005620      CLR   LAD
```

```

*****
:TEST 1:      EXERCISE MULF (MULTIPLY FLOATING)
:              ALL INTERRUPTS ON
:              ROUNDING MODE
*****

```

```

001224 104400          SCOPE
001226 012767 007400 177614  MOV      #007400, SFPS      ;SET IE BITS IN FORTRAN ANSWER
001234 005067 177612  CLR      SFEC              ;CLR FORTRAN FEC
001240 005067 177610  CLR      SFPC              ;CLR FORTRAN FPC
001244 005067 177572  CLR      FPS               ;CLR FPU FPS BUFFER
001250 005067 177570  CLR      FEC               ;CLR FPU FEC BUFFER
001254 005067 177566  CLR      FPC               ;CLR FPU FPC BUFFER
001260 004767 003624  JSR      PC,      RANDOM2 ;GET RANDOM INPUT DATA
001264 004467 003036  JSR      R4,      SPOLSH  ;ENTER POLISH MODE
001270 004340          SP.2A          ;PUSH 2 WORDS ON STACK (LONUM)
001272 004362          SP.2B          ;PUSH 2 WORDS ON STACK (HINUM)
001274 000000G        SMLR              ;ADDRESS OF FORTRAN MULTIPLY
001276 004430          STST              ;DETERMINE THE CONDITION CODES
001300 004374          SPOP2X          ;POP 2 WORDS AND EXIT POLISH MODE

001302 016700 177542  MOV      SFPS,      RO      ;DISPLAY FLOATING POINT STATUS
001306 170127 040000  LDFPS   #040000          ;CLEAR THE FPS, INTERRUPT DISABLE
001312 172467 177464  LDF     LONUM,  AC0        ;LOAD AC0 WITH A RANDOM NUMBER
001316 172567 177470  LDF     HINUM,  AC1        ;LOAD AC1 WITH A RANDOM NUMBER
001322 172767 177504  LDF     ANS2,   AC3        ;LOAD AC3 WITH THE SUM
001326 170127 007400  LDFPS   #007400          ;TURN INTERRUPTS ON
001332 012767 001340 005504  MOV     #.+6,  LAD        ;RESET LOOP ADDRESS

```

```

001340 172600          LDF     AC0,      AC2      ;LOAD AC0 INTO AC2
001342 171201          MULF   AC1,      AC2      ;MULTIPLY AC1 BY AC2
001344 005167 177500  RET1:  TST     SFPS        ;CHECK FOR FORTRAN FPS ERROR FLAG
001350 100412          BMI     ERR1          ;BRANCH IF ERROR FLAG SET
001352 170267 177464  STFPS   FPS                ;STORE FLOATING POINT STATUS
001356 026767 177460 177464  CMP     FPS,      SFPS     ;CHECK FPS
001364 001427          BEQ     TST1          ;BRANCH IF OK
001366 174267 177430  STF     AC2,      ANS1     ;SAVE FPU ANSWER
001372 104001          HLT+1          ;FPS ERROR
001374 000444          BR      END1          ;SKIP COMPARE

001376 170000          ERR1:  CFCC          ;WAIT FOR FPU TO FINISH
001400 026767 177436 177442  CMP     FPS, SFPS        ;CHECK THE FLOATING POINT STATUS
001406 001402          BEQ     .+6          ;BRANCH IF OK
001410 104377          HLT+377          ;FPS ERROR
001412 000435          BP      END1          ;SKIP TO END

001414 026767 177424 177430  CMP     FEC, SFEC        ;CHECK THE FLOATING EXCEPTION CODES
001422 001402          BEQ     .+6          ;BRANCH IF OK
001424 104377          HLT+377          ;FEC IS WRONG
001426 000427          BR      END1          ;SKIP TO END

001430 026767 177412 177416  CMP     FPC, SFPC        ;CHECK FLOATING PC
001436 001402          BEQ     TST1          ;BRANCH IF OK

```

```

001440 104377      HLT+377      ;WRONG ADDRESS IN FPC
001442 000421      BR          END1    ;SKIP TO END

001444 173702      TST1:  CHPF      AC2,   AC3    ;COMPARE FPU ANSWER TO FORTRAN ANSWER
001446 170000      CFCC          ;COPY FLOATING CONDITION CODES
001450 001416      BEQ          END1    ;ANSWERS CHECK
;COMPENSATE FOR FORTRAN INACCURACIES.
001452 174267      STF          AC2,   ANS1    ;SAVE FPU ANSWER
001456 162767      SUB          B1,    ANS1+2  ;DECREMENT FPU ANSWER
001464 005667      SBC          ANS1
001470 173767      CHPF      ANS1,   AC3    ;CHECK ANSWERS AGAIN
001474 170000      CFCC          ;COPY FLOATING CONDITION CODES
001476 001403      BEQ          END1    ;BRANCH IF OK
001500 174267      STF          AC2,   ANS1    ;SAVE FPU ANSWER
001504 104000      HLT          ;FPU AND FORTRAN DISAGREE

001506 005067      END1:  CLR          FPS      ;CLR FPU FPS BUFFER
001512 104400      SCOPE

```

```

;*****
;TEST 2:      EXERCISE MULD (MULTIPLY DOUBLE PRECISION)
;              ALL INTERRUPTS ON
;              ROUNDING MODE
;*****

```

```

001514 012767      MOV          #007600,SFPS ;SET IE BITS IN FORTRAN ANSWER
001522 005067      CLR          SFEC      ;CLR FORTRAN FEC
001526 007767      CLR          SFPC      ;CLR FORTRAN FPC
001530 007767      CLR          FPS      ;CLR FPU FPS BUFFER
001534 007767      CLR          FEC      ;CLR FPU FEC BUFFER
001538 005067      CLR          FPC      ;CLR FPU FPC BUFFER
001542 004767      JSR          PC,    RANDOM ;GET RANDOM INPUT DATA
001546 004467      JSR          R4,    SPOLSH ;ENTER POLISH MODE
001550 004330      SP. 48          ;PUSH 4 WORDS ON STACK (LONUM)
001554 004352      SP. 48          ;PUSH 4 WORDS ON STACK (HINUM)
001558 000000      MULD         ;ACCESS OF FORTRAN MULTIPLY
001562 004430      STST        ;DETERMINE THE CONDITION CODES
001566 004406      SPOP4X       ;POP 4 WORDS AND EXIT POLISH MODE

001570 016700      MOV          SFPS,    R0    ;DISPLAY FLOATING POINT STATUS
001574 170127      LDFPS      #04200      ;SET FD OF FPS ONLY, INTERRUPT DISABLE
001600 172467      LDD          LONUM,   AC0   ;LOAD AC0 WITH A RANDOM NUMBER
001604 172567      LDD          HINUM,   AC1   ;LOAD AC1 WITH A RANDOM NUMBER
001610 172767      LDD          ANS2,    AC3   ;LOAD AC3 WITH THE SUM
001614 170127      LDFPS      #007600      ;TURN INTERRUPTS ON
001620 012767      MOV          #.46,    LAD   ;RESET LOOP ADDRESS

```

```

;*****

```

```

001626 172600      RET2:  LDD          AC0,   AC2    ;LOAD AC0 INTO AC2
001630 171201      MULD         AC1,    AC2    ;MULTIPLY AC1 BY AC2
001632 005767      TST          SFPS      ;CHECK FOR FORTRAN FPS ERROR FLAG
001636 100412      BMI         ERR2      ;BRANCH IF ERROR FLAG SET
001640 170267      STFPS      FPS      ;STORE FLOATING POINT STATUS
001644 026767      CMP          FPS,    SFPS   ;CHECK FPS
001652 001427      BEQ          TST2     ;BRANCH IF OK

```

```

001654 174267 177142          STD      AC2,  ANS1      ;SAVE FPU ANSWER
001660 104001          HLT+1
001662 000450          BR        END2      ;FPS ERROR
                                ;SKIP TO END

001664 170000          ERR2:  CFCC
001666 026767 177150 177154  CMP      FPS, $FPS   ;WAIT FOR FPU TO FINISH
001674 001402          BEQ      .+6        ;CHECK THE FLOATING POINT STATUS
001676 104377          HLT+377
001700 000441          BR        END2      ;BRANCH IF OK
                                ;FPS ERROR
                                ;SKIP TO END

001702 026767 177136 177142  CMP      FEC, $FEC   ;CHECK THE FLOATING EXCEPTION CODES
001710 001402          BEQ      .+6        ;BRANCH IF OK
001712 104377          HLT+377
001714 000433          BR        END2      ;FEC IS WRONG
                                ;SKIP TO END

001716 026767 177124 177130  CMP      FPC, $FPC   ;CHECK FLOATING PC
001724 001402          BEQ      TST2      ;BRANCH IF OK
001726 104377          HLT+377
001730 000425          BR        END2      ;WRONG ADDRESS IN FPC
                                ;SKIP TO END

001732 173702          TST2:  CMPO     AC2,  AC3      ;COMPARE FPU ANSWER TO FORTRAN ANSWER
001734 170000          CFCC
001736 001422          BEQ      END2      ;COPY FLOATING CONDITION CODES
                                ;ANSWERS CHECK
                                ;COMPENSATE FOR FORTRAN INACCURACIES.
001740 174267 177056          STD      AC2,  ANS1      ;SAVE FPU ANSWER
001744 162767 000001 177056  SUB      #1,  ANS1+6  ;DECREMENT FPU ANSWER
001752 005667 177050          SBC     ANS1+4
001756 005667 177042          SBC     ANS1+2
001762 005667 177034          SBC     ANS1
001766 173767 177030          CMPO     ANS1,  AC3      ;CHECK ANSWERS AGAIN
001772 170000          CFCC
001774 001403          BEQ      END2      ;COPY FLOATING CONDITION CODES
                                ;BRANCH IF OK
001776 174267 177020          STD      AC2,  ANS1      ;SAVE FPU ANSWER
002002 104400          HLT
                                ;FPU AND FORTRAN DISAGREE

002004 005067 177032          END2:  CLR      FPS
002010 104400          SCOPE
                                ;CLR FPU FPS BUFFER

```

```

*****
:TEST 3:      EXERCISE MULF (MULTIPLY FLOATING)
:              OVERFLOW AND UNDERFLOW INTERRUPTS OFF.
:              ROUNDING MODE
*****

```

```

002012 012767 004400 177030  MOV      #004400, $FPS ;SET IE BITS IN FORTRAN ANSWER
002020 005067 177026          CLR      $FEC         ;CLR FORTRAN FEC
002024 005067 177024          CLR      $FPC         ;CLR FORTRAN FPC
002030 005067 177006          CLR      FPS          ;CLR FPU FPS BUFFER
002034 005067 177004          CLR      FEC          ;CLR FPU FEC BUFFER
002040 005067 177002          CLR      FPC         ;CLR FPU FPC BUFFER
002044 004767 003040          JSR     PC,  RANDOM2 ;GET RANDOM INPUT DATA
002050 004467 002252          JSR     R4,  $POLSH   ;ENTER POLISH MODE
002054 004340          SP, 29             ;PUSH 2 WORDS ON STACK (LONUM)
002056 004362          SP, 28             ;PUSH 2 WORDS ON STACK (MINUM)
002060 000000          $CLR
002062 004430          STST

```



```

002064 004374          SPOP2X          ;POP 2 WORDS AND EXIT POLISH MODE
002066 016700 176756      MOV      SFPS,  R0          ;DISPLAY FLOATING POINT STATUS
002072 170127 040000      LDFPS   #040000         ;CLEAR THE FPS, INTERRUPT DIS-ABLE
002076 172467 176700      LDF     LONUM,  AC0       ;LOAD AC0 WITH A RANDOM NUMBER
002102 172567 176704      LDF     HINUM,  AC1       ;LOAD AC1 WITH A RANDOM NUMBER
002106 172767 176720      LDF     ANS2,   AC3       ;LOAD AC3 WITH THE SUM
002112 170127 004400      LDFPS   #004400         ;TURN INTERRUPTS ON, EXCEPT OVERFLOW AND UNDERFLOW
002116 012767 002124 004720  MOV      #. +6,  LAD       ;RESET LOOP ADDRESS
    
```

```

002124 172600          LDF     AC0,   AC2       ;LOAD AC0 INTO AC2
002126 171201          MULF   AC1,   AC2       ;MULTIPLY AC1 BY AC2
002130 170267 176706      STFPS   FPS,          ;STORE FLOATING POINT STATUS
002134 026767 176702 176706  CMP     FPS,   SFPS      ;CHECK FPS
002142 001404          BEQ     TST3          ;BRANCH IF OK
002144 174267 176652      STF     AC2,   ANS1      ;SAVE FPU ANSWER
002150 104301          HLT+1  ;FPS ERROR
002152 000425          BR      END3          ;SKIP COMPARE
    
```

```

002154 032767 000002 176666  TST3:  BIT     #2,   SFPS      ;CHECK FOR OVERFLOW
002162 001021          BNE     EN3          ;BRANCH IF OVERFLOW
002164 173702          CMPP   AC2,   AC3       ;COMPARE FPU ANSWER TO FORTRAN ANSWER
002166 170000          CFCC          ;COPY FLOATING CONDITION CODES
002170 001416          BEQ     EN3          ;ANSWERS CHECK
; COMPENSATE FOR FORTRAN INACCURACIES.
002172 174267 176624          STF     AC2,   ANS1      ;SAVE FPU ANSWER
002176 162767 000001 176620  SUB     #1,   ANS1+2     ;DECREMENT FPU ANSWER
002204 005667 176612          SBC     ANS1,          ;CHECK ANSWERS AGAIN
002210 173767 176606          CMPP   ANS1,  AC3       ;COPY FLOATING CONDITION CODES
002214 170000          CFCC          ;BRANCH IF OK
002216 001403          BEQ     EN3          ;SAVE FPU ANSWER
002220 174267 176576          STF     AC2,   ANS1      ;FPU AND FORTRAN DISAGREE
002224 104000          HLT
    
```

```

002226 005067 176610          END3:  CLR     FPS          ;CLR FPU FPS BUFFER
002232 104400          SCOPE
    
```

; TEST 4: EXERCISE MULD (MULTIPLY DOUBLE PRECISION)
; OVERFLOW AND UNDERFLOW INTERRUPTS OFF
; ROUNDING MODE

```

002234 012767 004600 176606      MOV     #004600, SFPS    ;SET IE BITS IN FORTRAN ANSWER
002242 005067 176604          CLR     SFEC           ;CLR FORTRAN FEC
002246 005067 176602          CLR     SFPC           ;CLR FORTRAN FPC
002252 005067 176564          CLR     FPS           ;CLR FPU FPS BUFFER
002256 005067 176562          CLR     FEC           ;CLR FPU FEC BUFFER
002262 005067 176560          CLR     FPC           ;CLR FPU FPC BUFFER
002266 004767 002754          JSR     PC,   RANDOM    ;GET RANDOM INPUT DATA
002272 004467 002030          JSR     R4,   SPOLSH    ;ENTER POLISH MODE
002276 004330          SP. 4A          ;PUSH 4 WORDS ON STACK (LONUM)
002300 004352          SP. 4B          ;PUSH 4 WORDS ON STACK (HINUM)
002302 000000          SMLD          ;ADDRESS OF FORTRAN MULTIPLY
    
```

```

002304 004430          STST          ; DETERMINE THE CONDITION CODES
002306 004406          SPOP4X        ; POP 4 WORDS AND EXIT POLISH MODE

002310 016700 176534    MOV          SFPS          RO      ; DISPLAY FLOATING POINT STATUS
002314 170127 040200    LDFPS         8040200      ; SET FD OF FPS ONLY, INTERRUPT DISABLE
002320 172467 176456    LDO          LONUM,       AC0     ; LOAD AC0 WITH A RANDOM NUMBER
002324 172567 176462    LDO          MINUM,       AC1     ; LOAD AC1 WITH A RANDOM NUMBER
002330 172767 176478    LDO          ANS2,        AC3     ; LOAD AC3 WITH THE SUM
002334 170127 004600    LDFPS         8004600      ; TURN INTERRUPTS ON, EXCEPT OVER AND UNDERFLOW
002340 012767 002346 004476  MOV          8.+6,        LAD     ; RESET LOOP ADDRESS

```

```

002346 172600          LDO          AC0,         AC2     ; LOAD AC0 INTO AC2
002350 171201          MULD         AC1,         AC2     ; MULTIPLY AC1 BY AC2
002352 170267 176464    STFPS        FPS         ; STORE FLOATING POINT STATUS
002356 026767 176460 176464  CMP          FPS,         SFPS    ; CHECK FPS
002364 001404          BEQ          TST4         ; BRANCH IF OK
002366 174267 176430    STD          AC2,         ANS1    ; SAVE FPU ANSWER
002372 104001          HLT+1         ; FPU ERROR
002374 000431          BR          END4         ; SKIP COMPARE

002376 032767 000002 176444  TST4:      BIT          #2,         SFPS    ; CHECK FOR OVERFLOW
002404 001025          BNE          END4         ; BRANCH IF OVERFLOW
002406 173702          CMPO         AC2,         AC3     ; COMPARE FPU ANSWER TO FORTRAN ANSWER
002410 170000          CFCC        ; COPY FLOATING CONDITION CODES
002412 001422          BEQ          END4         ; ANSWERS CHECK
; COMPENSATE FOR FORTRAN INACCURACIES.
002414 174267 176402          STD          AC2,         ANS1    ; SAVE FPU ANSWER
002420 162767 000001 176402  SUB          #1,         ANS1+6  ; DECREMENT FPU ANSWER
002426 005667 176374          SBC          ANS1+4
002432 005667 176366          SBC          ANS1+2
002436 005667 176360          SBC          ANS1
002442 173767 176354          CMPO         ANS1,        AC3     ; CHECK ANSWERS AGAIN
002446 170000          CFCC        ; COPY FLOATING CONDITION CODES
002450 001403          BEQ          END4         ; BRANCH IF OK
002452 174267 176344          STD          AC2,         ANS1    ; SAVE FPU ANSWER
002456 104000          HLT         ; FPU AND FORTRAN DISAGREE

002460 005067 176356          END4:      CLR          FPS         ; CLR FPU FPS BUFFER
002464 104400          SCOPE

```

TEST 5: EXERCISE MULF (MULTIPLY FLOATING)
ALL INTERRUPTS ON
TRUNCATE MODE

002466	012767	007440	176354	MOV	0007440, SFPS	: SET IE BITS IN FORTRAN ANSWER
002474	005067	176352		CLR	SFEC	: CLR FORTRAN FEC
002500	005067	176350		CLR	SFPC	: CLR FORTRAN FPC
002504	005067	176332		CLR	FPS	: CLR FPU FPS BUFFER
002510	005067	176330		CLR	FEC	: CLR FPU FEC BUFFER
002514	005067	176326		CLR	FPC	: CLR FPU FPC BUFFER
002520	004767	002364		JSR	PC, R4,	: GET RANDOM INPUT DATA
002524	004467	001576		JSR	R4,	: ENTER POLISH MODE
002530	004340			SP. 29		: PUSH 2 WORDS ON STACK (LONUM)
002532	004362			SP. 28		: PUSH 2 WORDS ON STACK (HINUM)
002534	000000G			SHLR		: ADJ. 55 OF FORTRAN MULTIPLY
002536	004430			STST		: DETERMINE THE CONDITION CODES
002540	004374			SPOP2X		: POP 2 WORDS AND EXIT POLISH MODE
002542	016700	176302		MOV	SFPS, R0	: DISPLAY FLOATING POINT STATUS
002546	170127	040000		LDFPS	0040000	: CLEAR THE FPS, INTERRUPT DISABLE
002552	172467	176224		LDF	LONUM, AC0	: LOAD AC0 WITH A RANDOM NUMBER
002556	172567	176230		LDF	HINUM, AC1	: LOAD AC1 WITH A RANDOM NUMBER
002562	172767	176244		LDF	ANS2, AC3	: LOAD AC3 WITH THE SUM
002566	170127	007440		LDFPS	0007440	: TURN INTERRUPTS ON
002572	012767	002600	004244	MOV	0. +6, LAD	: RESET LOOP ADDRESS

002500	172600					
002502	171201					
002504	005767	176240		RETS:	LDF AC0, AC2	: LOAD AC0 INTO AC2
002510	100412				MULF AC1, AC2	: MULTIPLY AC1 BY AC2
002512	170267	176224			TST SFPS	: CHECK FOR FORTRAN FPS ERROR FLAG
002516	026767	176220	176224		BMI ERRS	: BRANCH IF ERROR FLAG SET
002524	001427				STFPS FPS	: STORE FLOATING POINT STATUS
002526	174267	176170			CMP FPS, SFPS	: CHECK FPS
002532	104001				BEQ TST5	: BRANCH IF OK
002534	000455				STF AC2, ANS1	: SAVE FPU ANSWER
					HLT+1	: FPS ERROR
					BR	: SKIP COMPARE
					ENDS	
002536	170000			ERRS:	CFCC	: WAIT FOR FPU TO FINISH
002540	026767	176176	176202		CMP FPS, SFPS	: CHECK THE FLOATING POINT STATUS
002546	001402				BEQ .+6	: BRANCH IF OK
002550	104377				HLT+377	: FPS ERROR
002552	000446				BR	: SKIP TO END
					ENDS	
002554	026767	176164	176170		CMP FEC, SFEC	: CHECK THE FLOATING EXCEPTION CODES
002562	001402				BEQ .+6	: BRANCH IF OK
002564	104377				HLT+377	: FEC IS WRONG
002566	000440				BR	: SKIP TO END
					ENDS	
002670	026767	176152	176156		CMP FPC, SFPC	: CHECK FLOATING PC
002676	001402				BEQ TST5	: BRANCH IF OK
002700	104377				HLT+377	: WRONG ADDRESS IN FPC

```

002702 000432          BR      ENDS          ;SKIP TO END
002704 173702          TST5:  CMPF    AC2,    AC3          ;COMPARE FPU ANSWER TO FORTRAN ANSWER
002706 170000          CFCC          ;COPY FLOATING CONDITION CODES
002710 001427          BEQ     ENDS          ;ANSWERS CHECK
          ;COMPENSATE FOR FORTRAN INACCURACIES.
002712 174267 176104  STF    AC2,    ANS1          ;SAVE FPU ANSWER
002716 062767 000001 176100  ADD    #1,    ANS1+2          ;INCREMENT FPU ANSWER
002724 005567 176072  ADC    ANS1          ;
002730 173767 176066  CMPF    ANS1,    AC3          ;CHECK ANSWERS AGAIN
002734 170000          CFCC          ;COPY FLOATING CONDITION CODES
002736 001414          BEQ     ENDS          ;BRANCH IF OK
002740 162767 000002 176056  SUB    #2,    ANS1+2          ;DECREMENT FPU ANSWER
002746 005567 176050  SBC    ANS1          ;
002752 173767 176044  CMPF    ANS1,    AC3          ;CHECK ANSWERS AGAIN
002756 170000          CFCC          ;COPY FLOATING CONDITION CODES
002760 001403          BEQ     ENDS          ;BRANCH IF OK
002762 174267 176034  STF    AC2,    ANS1          ;SAVE FPU ANSWER
002766 104000          HLT          ;FPU AND FORTRAN DISAGREE

002770 005067 176046  ENDS:  CLR    FPS          ;CLR FPU FPS BUFFER
002774 104400          SCOPE

```

```

*****
:TEST 6:      EXERCISE MULD (MULTIPLY DOUBLE PRECISION)
:              ALL INTERRUPTS ON
:              TRUNCATE MODE
*****

```

```

002776 012767 007640 176044  MOV    #007640, SFPS          ;SET IE BITS IN FORTRAN ANSWER
003004 005067 176042          CLR    SFEC          ;CLR FORTRAN FEC
003010 005067 176040          CLR    SFPC          ;CLR FORTRAN FPC
003014 005067 176022          CLR    FPS          ;CLR FPU FPS BUFFER
003020 005067 176020          CLR    FEC          ;CLR FPU FEC BUFFER
003024 005067 176016          CLR    FPC          ;CLR FPU FPC BUFFER
003030 004767 002212  JSR    PC,    RANDOM4          ;GET RANDOM INPUT DATA
003034 004467 001266  JSR    R4,    SPOLSH          ;ENTER POLISH MODE
003040 004330          SP.4A          ;POP 4 WORDS ON STACK (LONUM)
003042 004332          SP.4B          ;POP 4 WORDS ON STACK (MINUM)
003044 004330          $ _D          ;FINISH OF FORTRAN MULTIPLY
003046 004430          $1ST          ;LEAVE THE CONDITION CODES
003050 004406          SPOP4X          ;POP 4 WORDS AND EXIT POLISH MODE

003052 016700 175772          MOV    SFPS,    RD          ;DISPLAY FLOATING POINT STATUS
003056 170127 040200          LDFPS #040200          ;SET FD OF FPS ONLY, INTERRUPT DISABLE
003062 172467 175714          LDD    LONUM,    ACO          ;LOAD ACO WITH A RANDOM NUMBER
003066 172567 175720          LDD    MINUM,    AC1          ;LOAD AC1 WITH A RANDOM NUMBER
003072 172767 175734          LDD    ANS2,    AC3          ;LOAD AC3 WITH THE SUM
003076 170127 007640          LDFPS #007640          ;TURN INTERRUPTS ON
003102 012767 003110 003734  MOV    #.+6,    LAD          ;RESET LOOP ADDRESS

```

003110	172600									
003112	171201									
003114	005767	175730								
003120	100412									
003122	170267	175714								
003126	026767	175710	175714							
003134	001427									
003136	174267	175660								
003142	104001									
003144	000465									
RET6:										
				LDD	AC0,	AC2				:LOAD AC0 INTO AC2
				MULD	AC1,	AC2				:MULTIPLY AC1 BY AC2
				TST	SFPS					:CHECK FOR FORTRAN FPS ERROR FLAG
				BMI	ERR6					:BRANCH IF ERROR FLAG SET
				STFPS	FPS					:STORE FLOATING POINT STATUS
				CMP	FPS,	SFPS				:CHECK X FPS
				BEQ	TST6					:BRANCH IF OK
				STD	AC2,	ANS1				:SAVE FPU ANSWER
				HLT+1						:FPS ERROR
				BR	END6					:SKIP TO END
ERR6:										
				CFCC						:WAIT FOR FPU TO FINISH
				CMP	FPS,SFPS					:CHECK THE FLOATING POINT STATUS
				BEQ	.+6					:BRANCH IF OK
				HLT+377						:FPS ERROR
				BR	END6					:SKIP TO END
TST6:										
				CMPD	AC2,	AC3				:COMPARE FPU ANSWER TO FORTRAN ANSWER
				CFCC						:COPY FLOATING CONDITION CODES
				BEQ	END6					:ANSWERS CHECK
										:COMPENSATE FOR FORTRAN INACCURACIES.
				STD	AC2,	ANS1				:SAVE FPU ANSWER
				ADD	#1,	ANS1+6				:INCREMENT FPU ANSWER
				ADC	ANS1+4					
				ADC	ANS1+2					
				ADC	ANS1					
				CMPD	ANS1,	AC3				:CHECK ANSWERS AGAIN
				CFCC						:COPY FLOATING CONDITION CODES
				BEQ	END6					:BRANCH IF OK
				SUB	#2,	ANS1+6				:DECREMENT FPU ANSWER
				SBC	ANS1+4					
				SBC	ANS1+2					
				SBC	ANS1					
				CMPD	ANS1,	AC3				:CHECK ANSWERS AGAIN
				CFCC						:COPY FLOATING CONDITION CODES
				BEQ	END6					:BRANCH IF OK
				STD	AC2,	ANS1				:SAVE FPU ANSWER
				HLT						:FPU AND FORTRAN DISAGREE
END6:										
				CLR	FPS					:CLR FPU FPS BUFFER
				SCOPE						

: TEST 7: EXERCISE MULF (MULTIPLY FLOATING)
: OVERFLOW AND UNDERFLOW INTERRUPTS OFF.
: TRUNCATE MODE

:*****

003326	012767	004440	175514	MOV	#004440, SFPS	: SET IE BITS IN FORTRAN ANSWER
003334	005067	175512		CLR	SFEC	: CLR FORT M FEC
003340	005067	175510		CLR	SFPC	: CLR FORT M FPC
003344	005067	175472		CLR	FPS	: CLR FPU FPS BUFFER
003350	005067	175470		CLR	FEC	: CLR FPU FEC BUFFER
003354	005067	175466		CLR	FPC	: CLR FPU FPC BUFFER
003360	004767	001524		JSR	PC, RANDOM2	: GET RANDOM INPUT DATA
003364	004467	000736		JSR	R4, SPOLSH	: ENTER POLISH MODE
003370	004340			SP	.2A	: PUSH 2 WORDS ON STACK (LONUM)
003372	004362			SP	.2B	: PUSH 2 WORDS ON STACK (HINUM)
003374	000000G			S	P	: ADDRESS OF FORTRAN MULTIPLY
003376	004430			STST		: DETERMINE THE CONDITION CODES
003400	004374			SPOP2X		: POP 2 WORDS AND EXIT POLISH MODE
003402	016700	175442		MOV	SFPS, RO	: DISPLAY FLOATING POINT STATUS
003406	170127	040000		LDFPS	#040000	: CLEAR THE FPS, INTERRUPT DISABLE
003412	172467	175364		LDF	LONUM, ACO	: LOAD ACO WITH A RANDOM NUMBER
003416	172567	175370		LDF	HINUM, AC1	: LOAD AC1 WITH A RANDOM NUMBER
003422	172767	175404		LDF	R4, AC3	: LOAD AC3 WITH THE SUM
003426	170127	004440		LDFPS	#004440	: TURN INTERRUPTS ON, EXCEPT OVERFLOW AND UNDERFLOW
003432	012767	003440	003404	MOV	R.+6, LAD	: RESET LOOP ADDRESS

:*****

003440	172600			LDF	ACO, AC2	: LOAD ACO INTO AC2
003442	171201			MULF	AC1, AC2	: MULTIPLY AC1 BY AC2
003444	170267	175372		STFPS		: STORE FLOATING POINT STATUS
003450	026767	175366	175372	CMF	FPS, SFPS	: CHECK FPS
003456	001404			BEQ	TST7	: BRANCH IF OK
003460	174267	175336		STF	AC2, ANS1	: SAVE FPU ANSWER
003464	104001			HLT+1		: FPS ERROR
003466	000436			BR	END7	: SKIP COMPARE

003470	032767	000002	175352	TST7:	BIT	#2, SFPS	: CHECK FOR OVERFLOW
003476	001032				EIE	END7	: BRANCH IF OVERFLOW
003480	173702				CMF	AC2, AC3	: COMPARE FPU ANSWER TO FORTRAN ANSWER
003482	170000				CFCC		: COPY FLOATING CONDITION CODES
003484	001427				BEQ	END7	: ANSWERS CHECK

003506	174267	175310			: COMPENSATE FOR FORTRAN INACCURACIES.	
003512	062767	000001	175304	STF	AC2, ANS1	: SAVE FPU ANSWER
003520	005567	175276		ADD	#1, ANS1+2	: INCREMENT FPU ANSWER
003524	173767	175272		ADC	ANS1, AC3	: CHECK ANSWERS AGAIN
003530	170000			CMF	ANS1, AC3	: COPY FLOATING CONDITION CODES
003532	001414			CFCC		: BRANCH IF OK

003534	162767	000002	175262	BEQ	END7	: DECREMENT FPU ANSWER
003542	005667	175254		SUB	#2, ANS1+2	: CHECK ANSWERS AGAIN
003546	173767	175250		SBC	ANS1, AC3	: COPY FLOATING CONDITION CODES
003552	170000			CMF	ANS1, AC3	: BRANCH IF OK
003554	001403			CFCC		: SAVE FPU ANSWER
003556	174267	175240		BEQ	END7	
				STF	AC2, ANS1	

```

003562 104000          HLT          ;FPU AND FORTRAN DISAGREE
003564 005067 175252  END7:  CLR      FPS          ;CLR FPU FPS BUFFER
003570 104400          SCOPE

```

```

*****
:TEST 10:      EXERCISE MULD (MULTIPLY DOUBLE PRECISION)
:              OVERFLOW AND UNDERFLOW INTERRUPTS OFF
:              TRUNCATE MODE
*****

```

```

003572 012767 004640 175250  MOV      #104640, SFPS  ;SET IE BITS IN FORTRAN ANSWER
003600 005067 175246          CLR      $FEC          ;CLR FORTRAN FEC
003604 005067 175244          CLR      $FPC          ;CLR FORTRAN FPC
003610 005067 175226          CLR      FPS          ;CLR FPU FPS BUFFER
003614 005067 175224          CLR      FEC          ;CLR FPU FEC BUFFER
003620 005067 175222          CLR      FPC          ;CLR FPU FPC BUFFER
003624 004767 001416          JSR      PC,    RANDOM  ;GET RANDOM INPUT DATA
003630 004467 000472          JSR      R4,    SPOLSH ;ENTER POLISH MODE
003634 004330          SP. 4A          ;PUSH 4 WORDS ON STACK (LONUM)
003636 004352          SP. 4B          ;PUSH 4 WORDS ON STACK (HINUM)
003640 000000G          MULD          ;ADDRESS OF FORTRAN MULTIPLY
003642 004430          STST          ;DETERMINE THE CONDITION CODES
003644 004406          SPOP4X         ;POP 4 WORDS AND EXIT POLISH MODE

003646 016700 175176          MOV      SFPS,    R0    ;DISPLAY FLOATING POINT STATUS
003652 170127 040200          LDFPS   #040200      ;SET FD OF FPS ONLY, INTERRUPT DISABLE
003656 172467 175120          LDD     LONUM,    AC0  ;LOAD AC0 WITH A RANDOM NUMBER
003662 172567 175124          LDD     HINUM,    AC1  ;LOAD AC1 WITH A RANDOM NUMBER
003666 172767 175140          LDD     ANS2,     AC3  ;LOAD AC3 WITH THE SUM
003672 170127 004640          LDFPS   #004640      ;TURN INTERRUPTS ON, EXCEPT OVER AND UNDERFLOW
003676 012767 003704 003140  MOV      #. +6,    LAD  ;RESET LOOP ADDRESS

```

```

*****

```

```

003704 172600          LDD     AC0,    AC2    ;LOAD AC0 INTO AC2
003706 171201          RET10: MULD    AC1,    AC2  ;MULTIPLY AC1 BY AC2
003710 170267 175126          STFPS   FPS          ;STORE FLOATING POINT STATUS
003714 026767 175122 175126  CMP     FPS,    SFPS   ;CHECK FPS
003722 001404          BEQ     TST10        ;BRANCH IF OK
003724 174267 175072          STD     AC2,    ANS1  ;SAVE FPU ANSWER
003730 104001          HLT+1  ;FPS ERROR
003732 000446          BR     END10        ;SKIP COMPARE

003734 032767 000002 175106  TST10: BIT     #2,    SFPS  ;CHECK FOR OVERFLOW
003742 001042          BNE     END10        ;BRANCH IF OVERFLOW
003744 173702          CMPO   AC2,    AC3   ;COMPARE FPU ANSWER TO FORTRAN ANSWER
003746 170000          CFCC          ;COPY FLOATING CONDITION CODES
003750 001437          BEQ     END10        ;ANSWERS CHECK
:COMPENSATE FOR FORTRAN INACCURACIES.
003752 174267 175044          STD     AC2,    ANS1  ;SAVE FPU ANSWER
003756 062767 000001 175044  ADD     #1,    ANS1+6 ;INCREMENT FPU ANSWER
003764 005567 175036          RDC     ANS1+4
003770 005567 175030          RDC     ANS1+2
003774 005567 175022          RDC     ANS1
004000 173767 175016          CMPO   ANS1,    AC3   ;CHECK ANSWERS AGAIN

```

```

004004 170000          CFCC          ;COPY FLOATING CONDITION CODES
004006 001420          BEQ          END10      ;BRANCH IF OK
004010 162767 000002 175012  SUB          ANS1+6 ;DECREMENT FPU ANSWER
004016 005667 175004          SBC          ANS1+4
004022 005667 171776          SBC          ANS1+2
004026 005667 174770          SBC          ANS1
004032 173767 174764  CMPD         ANS1,  AC3 ;CHECK ANSWERS AGAIN
004036 170000          CFCC          ;COPY FLOATING CONDITION CODES
004040 001403          BEQ          END10      ;BRANCH IF OK
004042 174267 174754          STD          ANS1 ;SAVE FPU ANSWER
004046 104000          HLT ;FPU AND FORTRAN DISAGREE

004050 005067 :174766          END10: CLR          FPS ;CLR FPU FPS BUFFER
004054 104400          SCOPE

```



```

004056 032737 002000 177570 DONE: BIT      @SW10,@SWR      ;RING THE BELL?
004064 001005                BNE      1$          ;NO!
004066 012767 000207 002744      MOV      @BELL,.TYPE ;TYPE A BELL
004074 001004 007040                TYPE,    .TYPE
004100 001046                CLR      -(6)        ;CLEAR TRACE TRAP
004102 032737 010000 177570 1$:      BIT      @SW12,@SWR  ;RUN WITH TRT?
004110 001010                BNE      2$          ;
004112 005167 002724                COM      TRPB
004116 100005                BPL      2$
004120 052716 000020                BIS      @20,(6)    ;SET TRACE TRAP
004124 012746 004160                MOV      @HERE,-(6) ;JUMP TO START OF TEST
004130 000002                RTI
004132 012746 004140                2$:      MOV      @4$, -(6) ;JUMP TO START OF TEST
004136 000002                RTI
004140 013700 000042                4$:      MOV      @#42,R0  ;GET MONITOR ADDRESS
004144 001405                BEQ     HERE        ;IF NONE
004146 000005                RESET
004150 004710                SENDAD: JSR     PC,(0) ;GO TO MONITOR
004152 000240                NOP
004154 000240                NOP
004156 000240                NOP
004160 000137 000200      HERE:  JMP     @#200    ;JUMP TO START OF TEST
004164 000002      YESRT: RTI        ;RETURN TO PROGRAM FROM TRAP

004166 032737 000400 177570 .ENT:  BIT      @SW08,@SWR  ;KILL LDUB OR LOOP ON SPEC. TEST
004174 001404                BEQ     1$
004176 123767 177570 174574      CMPB   @SWR,ICNT    ;ON RIGHT TEST? *SW7-0*
004204 001437                BEQ     OVER
004206 113703 177570                1$:      MOVB   @SWR,R3     ;GET UB BITS
004212 170003                LDUB
004214 032737 040000 177570      BIT      @SW14,@SWR ;LOOP ON TEST
004222 001026                BNE     KIT
004224 032737 004000 77570      BIT      @SW11,@SWR ;KILL ITERATIONS
004232 001012                BNE     SAVLAD
004234 105767 174541                TSTB   ICNT+1
004240 001404                BEQ     2$          ;BRANCH IF FIRST
004242 126767 002604 174531      CMPB   TIMES,ICNT+1 ;DONE?
004250 001013                BNE     KIT        ;BRANCH IF NOT
004252 112767 000001 174521 2$:      MOVB   #1,ICNT+1   ;FIRST ITERATION
004260 105267 174514      SAVLAD: INCB   ICNT   ;COUNT TEST NUMBERS
004264 011667 002554                MOV     (6),LAD    ;SAVE LOOP ADDRESS
004270 016737 174504 177570      MOV     ICNT,@#DISPLAY ;DISPLAY TEST NO. AND ITERATION COUNT
004276 000002                RTI                ;RETURN

004300 105267 174475                KIT:     INCB   ICNT+1
004304 016737 174470 177570      OVER:  MOV     ICNT,@#DISPLAY ;SET UP DISPLAY
004312 005767 002526                TST    LAD        ;FIRST ONE?
004316 001760                BEQ    SAVLAD
004320 016716 002520                MOV     LAD,(6)   ;FUDGE RETURN ADDRESS
004324 000002                RTI                ;FIXES PS

```

```

004326 000134          SFOLSH: JMP      2(R4)+
004330 016746 174454    SP.4A: MOV      LONUM+6,-(SP)
004334 016746 174446    MOV      LONUM+4,-(SP)
004340 016746 174440    SP.2A: MOV      LONUM+2,-(SP)
004344 016746 174432    MOV      LONUM,-(SP)
004350 000134          JMP      2(R4)+
004352 016746 174442    SP.4B: MOV      HINUM+6,-(SP)
004357 016746 174434    MOV      HINUM+4,-(SP)
004362 016746 174426    SP.2B: MOV      HINUM+2,-(SP)
004366 016746 174420    MOV      HINUM,-(SP)
004372 000134          JMP      2(R4)+
004374 012667 174432    SPOP2X: MOV     (SP)+, ANS2
004400 012667 174430    MOV     (SP)+, ANS2+2
004404 000204          RTS      R4          ;EXIT POLISH MODE
004406 012667 174420    SPOP4X: MOV     (SP)+, ANS2
004412 012667 174416    MOV     (SP)+, ANS2+2
004416 012667 174414    MOV     (SP)+, ANS2+4
004422 012667 174412    MOV     (SP)+, ANS2+6
004426 000204          RTS      R4          ;EXIT POLISH MODE
004430 032767 100002 174412 STST:  BIT      #100002,$FPS ;CHECK FOR ERROR FLAG AND OVERFLOW
004436 100415          BMI      STER ;BRANCH IF FLAG SET
004440 001023          BNE      STOV ;BRANCH IF OVERFLOW
004442 005716          TST     (6) ;FIND THE SIGN
004444 100003          BPL     .+10 ;BRANCH IF PLUS
004446 052767 000010 174374  BIS     #10, $FPS ;SET N BIT
004454 032716 077600          BIT     #077600,(6) ;TEST THE EXPONENT
004460 001003          BNE      .+10 ;BRANCH IF NOT ZERO
004462 052767 000004 174360  BIS     #04, $FPS ;SET Z BIT
004470 000134          JMP      2(R4)+
004472 116701 174302    STER:  MOV8    ICNT, R1 ;GET TEST NUMBER
004476 005301          DEC     R1 ;SET POINTER
004500 006301          ASL    R1 ;TEST # * 2
004502 016167 004526 174344  MOV     RETAD(1), $FPC ;STORE FORTRAN FPC
004510 022626          STOV:  CMP     (SP)+, (SP)+ ;"POP" 2 WORDS
004512 105767 174332    TSTB   $FPS ;CHECK FD BIT
004516 100001          BPL     STOV1 ;BRANCH IF FLOATING MODE
004520 022626          CMP     (SP)+, (SP)+ ;"POP" 2 MORE WORDS
004522 005724          STOV1: TST     (R4)+ ;SKIP "POPX" ROUTINE
004524 000204          RTS     R4 ;EXIT POLISH MODE
004526 001342    RETAD: RET1
004530 001630          RET2
004532 000000          DUMMY
004534 000000          DUMMY
004536 002602          RET5
004540 003112          RET6
004542 000000          DUMMY
004544 000000          DUMMY

```

004546	020027	003405		\$ERRA:	CMP	RO,	#3405		:CHECK FOR UNDERFLOW, FD=0
004552	001412				BEQ	SUNDR0			:BRANCH IF UNDERFLOW
004554	020027	003005			CMP	RO,	#3005		:CHECK FOR UNDERFLOW, FD=1
004560	001425				BEQ	SUNDR1			:BRANCH IF UNDERFLOW
004562	020027	006003			CMP	RO,	#6003		:CHECK FOR OVERFLOW, FD=0
004566	001462				BEQ	\$OVER0			:BRANCH IF OVERFLOW
004570	020027	005003			CMP	RO,	#5003		:CHECK FOR OVERFLOW, FD=1
004574	001471				BEQ	\$OVER1			:BRANCH IF OVERFLOW
004576	000000			\$UNKER:	HALT				:UNKNOWN ERROR!
004600	032767	003000	174242	\$UNDR0:	BIT		#003000,	\$FPS	:CHECK FOR INTERRUPTS ON OR OFF
004606	001512				BEQ	\$ERTS			:BRANCH IF OFF
004610	004467	177512			JSR	R4,	\$POLSH		:ENTER POLISH MODE
004614	004340				SP.2A				:PUSH 2 WORDS ON STACK (LONUM)
004616	004362				SP.2B				:PUSH 2 WORDS ON STACK (HINUM)
004620	004704				\$A0200				:ADD 200 TO ONE EXPONENT
004622	000000G				\$MLR				:ADDRESS OF FORTRAN MULTIPLY
004624	004712				\$2005B				:SUBTRACT 200 FROM THE EXPONENT OF ANS
004626	004430				\$TST				:DETERMINE CONDITION CODES
004630	004374				\$POP2X				:POP 2 WORDS AND EXIT POLISH MODE
004632	000415				BR	\$UNDR0			
004634	032767	003000	174206	\$SUNDR1:	BIT		#003000,	\$FPS	:CHECK FOR INTERRUPTS ON OR OFF
004642	001474				BEQ	\$ERTS			:BRANCH IF OFF
004644	004467	177456			JSR	R4,	\$POLSH		:ENTER POLISH MODE
004650	004330				SP.4A				:PUSH 2 WORDS ON STACK (LONUM)
004652	004352				SP.4B				:PUSH 2 WORDS ON STACK (HINUM)
004654	004704				\$A0200				:ADD 200 TO ONE EXPONENT
004656	000000G				\$MLD				:ADDRESS OF FORTRAN MULTIPLY
004660	004712				\$2005B				:SUBTRACT 200 FROM THE EXPONENT OF ANS
004662	004430				\$TST				:DETERMINE CONDITION CODES
004664	004406				\$POP4X				:POP 4 WORDS AND EXIT POLISH MODE
004666	052767	100000	174154	\$SUNDR0:	BIS		#100000,	\$FPS	:SET FPS ERROR FLAG
004674	012767	000012	174150		MOV	#12,	\$FEC		:SET UNDERFLOW EXCEPTION CODE
004702	000205				RTS	%5			:RETURN TO FORTRAN ROUTINE
004704	062716	040000		\$A0200:	ADD		#040000,	(SP)	
004710	000134				JMP	2(R4)+			
004712	162716	040000		\$2005B:	SUB		#040000,	(SP)	
004716	032767	000003	173052		BIT	#3,	PS		:TEST FOR C OR V BITS
004724	001402				BEQ	\$2015B			
004726	062716	100000		\$2015B:	ADD		#100000,	(SP)	
004732	000134				JMP	2(R4)+			
004734	004467	177366		\$OVER0:	JSR	R4,	\$POLSH		:ENTER POLISH MODE
004740	004340				SP.2A				:PUSH 2 WORDS ON STACK (LONUM)
004742	004362				SP.2B				:PUSH 2 WORDS ON STACK (HINUM)
004744	005036				\$S8200				:SUBTRACT 200 FROM ONE EXPONENT
004746	000000G				\$MLR				:ADDRESS OF FORTRAN MULTIPLY
004750	005044				\$200A0				:ADD 200 TO THE EXPONENT OF ANS
004752	004430				\$TST				:DETERMINE CONDITION CODES
004754	004374				\$POP2X				:POP 2 WORDS AND EXIT POLISH MODE
004756	000411				BR	\$OVER0			
004760	004467	177342		\$OVER1:	JSR	R4,	\$POLSH		:ENTER POLISH MODE

MAINDEC-11-DCFPT-0
DCFPTC.P11

FLOATING POINT MULTIPLY EXERCISER
FORTRAN ROUTINES (POLISH MODE)

```

004764 004330      SP. 4A
004766 004352      SP. 4B
004770 005036      $$S8200
004772 000000G     $MLO
004774 005044      $200A0
004776 004430      $TST
005000 004406      $POPA4X
005002 032767 003000 174040 $OVERA: BIT      8003000, $FPS
005010 001406      BEQ      $OVR1
005012 052767 100000 174030  BIT      8100000, $FPS
005020 012767 000010 174024  MOV      810, $FEC
005026 052767 000002 174014 $OVR1: BIT      802, $FPS
005034 000205      SERTS: RTS      XS

005036 162716 040000      $$S8200: SUB      8040000, (SP)
005042 000134      JMP      2(R4)+

005044 052716 140000      $200A0: ADD      8140000, (SP)
005050 000134      JMP      2(R4)+

```

```

: PUSH 2 WORDS ON STACK (LONUM)
: PUSH 2 WORDS ON STACK (HINUM)
: SUBTRACT 200 FROM ONE EXPONENT
: ADDRESS OF FORTRAN MULTIPLY
: ADD 200 TO THE EXPONENT OF ANS
: DETERMINE CONDITION CODES
: POP 4 WORDS AND EXIT POLISH MODE
: CHECK FOR INTERRUPTS ON OR OFF
: BRANCH IF OFF
: SET FPS ERROR FLAG
: SET OVERFLOW EXCEPTION CODE
: SET OVERFLOW BIT IN FPS

```

```

: *****
: FPP INTERRUPT SERVICE ROUTINE
: *****

```

```

005052 170267 173764      FLTERR: STFPS   FPS
005056 170367 173762      STST   FEC
005062 032767 040000 173760  BIT      840000, $FPS
005070 001402      BEQ      .+6
005072 104377      HLT+377
005074 000404      BR      ERRTI

005076 005767 173746      TST      $FPS
005102 100401      BMI     ERRTI
005104 104377      HLT+377

005106 000002      ERRTI: RTI

```

```

: STORE FLOATING POINT STATUS
: STORE FLOATING EXCEPTION CODES
: CHECK INTERRUPT DISABLE
: BRANCH IF OFF
: INTERRUPT NOT SUPPOSED TO BE ENABLED

: CHECK FORTRAN FPS FOR ERROR FLAG
: BRANCH IF FLAG SET
: FLOATING POINT STATUS ERROR

```

005110	010046		RANDM2: MOV	X0, -(6)	SAVE R0
005112	010146		MOV	X1, -(6)	SAVE R1
005114	010246		MOV	X2, -(6)	SAVE R2
005116	010346		MOV	X3, -(6)	SAVE R3
005120	010446		MOV	X4, -(6)	SAVE R4
005122	010546		MOV	X5, -(6)	SAVE R5
005124	012704	001002	MOV	#LONUM, X4	SET UP LONUM POINTER
005130	012705	001012	MOV	#HINUM, X5	SET UP HINUM POINTER
005134	011400		MOV	(4), X0	SET R0 WITH LOW
005136	011501		MOV	(5), X1	SET R1 WITH HIGH
005140	012703	177771	REPET2: MOV	#-7, X3	SET SHIFT COUNT
005144	005002		SHIFT2: CLR	X2	
005146	076300		ASL	X0	SHIFT R0 LEFT AND
005148	06101		ROL	X1	ROTATE CARRY INTO R1 AND
005150	006102		ROL	X2	ROTATE CARRY INTO R2
005154	005203		INC	X3	CHECK FOR DONE
005156	001373		BNE	SHIFT2	CONTINUE SHIFT LOOP
005160	051402		ADD	(4), X2	ADD NUMBER TO MAKE X 129
005162	055501		RDC	X1	PROPOGATE CARRY
005164	061501		ADD	(5), X1	ADD NUMBER TO MAKE X 129
005166	005502		RDC	X2	PROPOGATE CARRY
005170	02700	001057	ADD	#1057, X0	ADD LOW CONSTANT
005174	05501		RDC	X1	PROPOGATE CARRY
005176	05502		RDC	X2	PROPOGATE CARRY
005200	062701	047401	ADD	#47401, X1	ADD HIGH CONSTANT
005204	05502		RDC	X2	PROPOGATE CARRY
005206	052702	000006	ADD	#6, X2	ADD HIGHEST CONSTART
005212	060200		ADD	X2, X0	REPRIME R0 WITH HIGHEST DIGIT
005214	005501		RDC	X1	PROPOGATE CARRY
005216	010024		MOV	X0, (4)+	SAVE R0
005220	010125		MOV	X1, (5)+	SAVE R1
005222	020427	001006	CMP	X4, #LONUM+4	CHECK FOR DONE ENOUGH
005226	001344		BNE	REPET2	BRANCH IF NOT DONE
005230	012605		MOV	(6)+, X5	RESTORE R5
005232	012604		MOV	(6)+, X4	RESTORE R4
005234	012603		MOV	(6)+, X3	RESTORE R3
005236	012602		MOV	(6)+, X2	RESTORE R2
005240	012601		MOV	(6)+, X1	RESTORE R1
005242	012600		MOV	(6)+, X0	RESTORE R0
005244	000207		RTS	X7	RETURN

005246	010046		RANDM4:	MOV	X0, -(6)	SAVE R0
005248	010146			MOV	X1, -(6)	SAVE R1
005250	010246			MOV	X2, -(6)	SAVE R2
005252	010346			MOV	X3, -(6)	SAVE R3
005254	010446			MOV	X4, -(6)	SAVE R4
005256	010546			MOV	X5, -(6)	SAVE R5
005262	012704	001002		MOV	#LONUM, X4	SET UP LONUM POINTER
005266	012705	001012		MOV	#HINUM, X5	SET UP HINUM POINTER
005272	011400			MOV	(4), X0	SET R0 WITH LOW
005274	011501			MOV	(5), X1	SET R1 WITH HIGH
005276	012703	177771	REPET4:	MOV	#-7, X3	SET SHIFT COUNT
005302	005002			CLR	X2	
005304	006300		SHIFT4:	ASL	X0	SHIFT R0 LEFT AND
005306	006101			ROL	X1	ROTATE CARRY INTO R1 AND
005310	006102			ROL	X2	ROTATE CARRY INTO R2
005312	005203			INC	X3	CHECK FOR DONE
005314	001373			BNE	SHIFT4	CONTINUE SHIFT LOOP
005316	061402			ADD	(4), X2	ADD NUMBER TO MAKE X 129
005320	005501			RDC	X1	PROPAGATE CARRY
005322	061501			RDC	(5), X1	ADD NUMBER TO MAKE X 129
005324	005502			RDC	X2	PROPAGATE CARRY
005326	062700	001057		RDC	#1057, X0	ADD LOW CONSTANT
005328	005501			RDC	X1	PROPAGATE CARRY
005330	005502			RDC	X2	PROPAGATE CARRY
005332	062701	047401		RDC	#47401, X1	ADD HIGH CONSTANT
005334	005502			RDC	X0	PROPAGATE CARRY
005336	062702	000006		RDC	#6, X2	ADD HIGHEST CONSTANT
005338	060200			RDC	X2, X0	REPRIME R0 WITH HIGHEST DIGIT
005340	005501			RDC	X1	PROPAGATE CARRY
005342	010024			MOV	X0, (4)+	SAVE R0
005344	010125			MOV	X1, (5)+	SAVE R1
005346	020427	001012		CMF	X4, #LONUM+10	CHECK FOR DONE ENOUGH
005348	001344			BNE	REPET4	BRANCH IF NOT DONE
005366	012605			MOV	(6)+, X5	RESTORE R5
005370	012604			MOV	(6)+, X4	RESTORE R4
005372	012603			MOV	(6)+, X3	RESTORE R3
005374	012602			MOV	(6)+, X2	RESTORE R2
005376	012601			MOV	(6)+, X1	RESTORE R1
005400	012600			MOV	(6)+, X0	RESTORE R0
005402	000207			RTS	X7	RETURN

```

005404 032767 002000 172156 .TRP: BIT #2000,SMR
005412 001405 BEQ .ET
005414 012767 000207 001416 MOV #BELL,.TYPE ;TYPE A BELL
005422 C 0004 007040 TYPE, TYPE
0 54 6 C 67 001414 .ET: INC ERRORS ;COUNT THE NUMBER OF ERRORS
005432 0 767 020000 172130 BIT #20000,SMR ;SKIP TYPEOUT IF SET
005440 001116 BNE NHEAD
005442 174767 173354 STF AC2, ANS1 ;SAVE THE ANSWER TO BE SURE
005446 000134 006744 TYPE, RET
005452 000004 006744 TYPE, RET
005456 011646 MOV (6) -(6) ;PUT ADDRESS OF INSTRUCTION ON STACK
005460 162716 000002 SUB #2,(6)
005464 117667 000000 001356 MOVB @6), WORDS
005472 012705 MOV (6)+,TTY ;TYPE (6)+ IN OCTAL
005474 004767 000546 JSR X7,PRINTR ;TYPE LEADING ZERO'S
00 00 000704 006741 TYPE, SPACE+3
00 04 012703 001002 MOV #LONUM, X3 ;SET UP POINTER
005510 004767 000174 JSR 7, STYPE ;TYPE A FLOATING POINT NUMBER
005514 010004 006747 TYPE, $SIGN
005520 0 4767 000164 JSR 7, STYPE ;TYPE A FLOATING POINT NUMBER
005524 0 04 0 5772 TYPE, FPUKN
005528 0 1767 0 0154 JSR 7, STYPE ;TYPE A FLOATING POINT NUMBER
005532 0 0004 0 6740 TYPE, SPACE+2
005536 0 5705 173276 MOV FPS,TTY ;TYPE FPS IN OCTAL
005540 0 4767 0 0476 JSR X7,PRINTR ;TYPE LEADING ZERO'S
005544 105767 0 1274 TSTB WORDS ;CHECK FOR STATUS ERROR
005548 100014 BPL .STAT ;BRANCH IF NOT
005552 0 04 006740 TYPE, SPACE+2
005556 016705 173256 MOV FEC,TTY ;TYPE FEC IN OCTAL
005560 004767 0 0434 JSR X7,PRINTR ;TYPE LEADING ZERO'S
005564 000004 0 6740 TYPE, SPACE+2
005568 016705 173244 MOV FPC,TTY ;TYPE FPC IN OCTAL
005572 004767 0 0440 JSR X7,PRINTR ;TYPE LEADING ZERO'S
005576 0 004 0 5753 .STAT: TYPE, FORTAN
005580 004767 0 0072 JSR 7, STYPE ;TYPE A FLOATING POINT NUMBER
005584 105767 001226 NOHEAD: TSTB WORDS
005588 001425 BEQ NHEAD
005592 000004 006740 TYPE, SPACE+2
005596 016705 173214 MOV SFPS,TTY ;TYPE SFPS IN OCTAL
005600 004767 000406 JSR X7,PRINTR ;TYPE LEADING ZERO'S
005604 105767 001204 TSTB WORDS
005608 100014 BPL NHEAD
005612 000004 006740 TYPE, SPACE+2
005616 016705 173174 MOV SFEC,TTY ;TYPE SFEC IN OCTAL
005620 004767 000364 JSR X7,PRINTR ;TYPE LEADING ZERO'S
005624 000004 006740 TYPE, SPACE+2
005628 016705 173162 MOV SFPC,TTY ;TYPE SFPC IN OCTAL
005632 004767 000350 JSR X7,PRINTR ;TYPE LEADING ZERO'S
005636 005737 177570 NHEAD: TST @#SMR
005700 100001 BPL .+4
005704 000000 HALT
005708 000002 RTI

```

F03

MAINDEC-11-DCFPT-D
DCFPT.E.P11

FLOATING POINT MULTIPLY EXERCISER
FLOATING POINT TYPEOUT SUBROUTINE

MACY11 27(732) 17-SEP-76 10:19 PAGE 31

```

005710 032767 001000 171652 STYPE: BIT      #1000, SWR      ;CHECK TTY FORMAT
0 5716 901007                                BNE      TYPE1
005720 105767 173124                                TSTB    $FPS
005724 100432                                BMI      TYPED
005726 004767 000076                                JSR     7, TYPEF
005732 022323                                TYPEA:  CMP    (3)+, (3)+ ;UP DATE THE TYPEOUT POINTER
005734 000207                                RTS     7

005736                                TYPE1:
005736 012305                                MOV     (3)+, TTY          ;TYPE (3)+ IN OCTAL
005740 004767 000302                                JSR    x7, PRINTR        ;TYPE LEADING ZERO'S
0 5744 00074 006742                                TYPE,   SPACE+4
0 5750 012305                                MOV     (3)+, TTY          ;TYPE (3)+ IN OCTAL
005752 004767 000270                                JSR    x7, PRINTR        ;TYPE LEADING ZERO'S
005756 105767 173066                                TSTB    $FPS
005762 100753                                BPL     TYPEA
005764 00074 006742                                TYPE,   SPACE+4
005770 012305                                MOV     (3)+, TTY          ;TYPE (3)+ IN OCTAL
005772 004767 000250                                JSR    x7, PRINTR        ;TYPE LEADING ZERO'S
005776 000004 006742                                TYPE,   SPACE+4
006002 012305                                MOV     (3)+, TTY          ;TYPE (3)+ IN OCTAL
006004 004767 000236                                JSR    x7, PRINTR        ;TYPE LEADING ZERO'S
006010 000207                                RTS     7

006012 012346                                TYPED: MOV    (3)+, -(6)      ;GET WORD 1
006014 012346                                MOV    (3)+, -(6)      ;GET WORD 2
006016 012346                                MOV    (3)+, -(6)      ;GET WORD 3
006020 012346                                MOV    (3)+, -(6)      ;GET WORD 4
006022 012746 000022                                MOV    #18, -(6)       ;CHAR COUNT
006026 000406                                BR     TYPE1
006030 012346                                TYPEF: MOV    (3)+, -(6)      ;GET WORD 1
006032 012346                                MOV    (3)+, -(6)      ;GET WORD 2
006034 000746                                CLR    -(6)            ;CLEAR WORD 3
006036 000746                                CLR    -(6)
006040 012746 000010                                MOV    #8, -(6)        ;CHAR COUNT
006044 004767 000104                                TYPE1: JSR    7, TY1         ;TYPE 1 BIT
006050 105766 000011                                TSTB   9, (6)          ;CHECK EXPONENT
006054 001001                                BNE    .+4             ;BRANCH ON NON-ZERO EXPONENT
006056 005116                                COM    (6)             ;FLAG ZERO EXPONENT
006060 000004 006742                                TYPE,   SPACE+4
006064 004767 000072                                JSR    7, TY2          ;TYPE 2 BITS
006070 004767 000074                                JSR    7, TY3          ;TYPE 3 BITS
006074 004767 00007C                                JSR    7, TY3          ;TYPE 3 BITS
006100 000004 006742                                TYPE,   SPACE+4
006104 004767 0000E0                                JSR    7, TYH          ;TYPE 2 BITS
006110 004767 000051                                TYPE2: JSR    7, TY3          ;TYPE 3 BITS
006114 005316                                DEC    (6)             ;DONE?
006116 001374                                BNE    TYPE2
006120 022626                                CMP    (6)+, (6)+      ;RESTORE
006122 022626                                CMP    (6)+, (6)+      ;THE
006124 005726                                TST    (6)+            ;STACK
006126 000207                                RTS     7

```


006130	005766	000002		TYH:	TST	2(6)		:CHECK FOR ZERO EXPONENT FLAG
006134	100405				BMI	TYZ		:BRANCH ON ZERO EXPONENT
006136	012746	000001			MOV	#1, -(6)		:TYPE HIDDEN BIT AND ONE
006142	011667	000672			MOV	(6), .TYPE		:FUDGE HIDDEN BIT
006146	000414				BR	TY+4		
006150	005166	000002		TYZ:	COM	2(6)		:GET RID OF ZERO EXPONENT FLAG
006154	012746	000001		TY1:	MOV	#1, -(6)		:TYPE 1 BIT
006160	000405				BR	TY		
006162	012746	000002		TY2:	MOV	#2, -(6)		:TYPE 2 BITS
006166	000402				BR	TY		
006170	012746	000003		TY3:	MOV	#3, -(6)		:TYPE 3 BITS
006174	005067	000640		TY:	CLR	.TYPE		
006200	016166	000006			ROL	6(6)		:SHIFT WORD 4
006204	016166	000010			ROL	8(6)		:SHIFT WORD 3
006210	006166	000012			ROL	10(6)		:SHIFT WORD 2
006214	006166	000014			ROL	12(6)		:SHIFT WORD 1
006220	016167	000614			ROL	.TYPE		:GET IT
006224	015316				DEC	(6)		:DONE?
006226	001364				BNE	TY+4		
006230	052767	000060	000602		BIS	#0, .TYPE		:MAKE IT ASCII
006236	000004	007040			TYPE,	.TYPE		:TYPE IT
006242	005726				TST	(6)+		:RESTORE THE STACK
006244	000207				RTS	7		

```

006246 112767 000001 000130 PRINTR: MOVB #1, .PR ;SET ZERO FILL SWITCH
006254 000402 BR .+6
006256 005767 000122 PRINTS: CLR .PR ;SUPPRESS LEADING ZERO'S
006258 112767 177772 000115 MOVB #6, .PR+1 ;SET COUNT
006260 010446 MOV R4, -(6) ;SAVE R4
006272 012704 006374 MOV #38, R4 ;SET POINTER TO FIRST ASCII CHAR.
006276 105014 CLRB (4) ;CLEAR FIRST BYTE
006280 000405 BR 28 ;ROTATE FIRST BIT
006284 105014 18: CLRB (4) ;CLEAR BYTE OF CHARACTER
006288 006105 ROL ITY ;ROTATE BIT INTO C
006292 106114 ROLB (4) ;PACK IT
006296 006105 ROL ITY ;ROTATE BIT INTO C
006300 106114 ROLB (4) ;PACK IT
006304 006105 28: ROL ITY ;ROTATE BIT INTO C
006308 106114 ROLB (4) ;PACK IT
006312 106114 ROL ITY ;ROTATE BIT INTO C
006316 106114 ROLB (4) ;PACK IT
006320 105714 TSTB (4)
006324 001402 BEQ .+6
006328 107767 000054 INCB .PR
006332 105767 000050 TSTB .PR ;CHECK FILL SWITCH
006336 001402 BEQ .+6
006340 152724 000060 BISB #'0, (4)+ ;MAKE INTO ASCII CHAR
006344 107767 000037 INCB .PR+1
006348 001355 BNE 18 ;REPEAT
006352 022704 006374 CMP #38, R4
006356 001002 BNE .+6
006360 112724 020060 MOVB #'0, (4)+
006364 105014 CLRB (4)
006368 000004 006174 TYPE 38 ;TYPE IT
006372 000207 MOV (6)+, R4 ;RESTORE R4
RTS PC

006374 000004 38: .BLKW 4
006404 000000 .PR: 0

006406 005267 000434 ERROR: INC ERRORS ;COUNT ERRORS
006412 132737 000001 000041 BITB #1, #41 ;AUTO MODE?
006420 001412 BEQ 18 ;NO!
006424 022767 000010 000416 CMP #10, ERRORS ;TOO MANY?
006430 001006 BNE 18 ;NOT YET
006432 013700 000042 MOV #42, R0 ;GET ADDRESS
006436 001403 BEQ 18 ;FORGET IT IF ZERO
006440 005037 000042 CLR #42 ;ZAP 42
006444 004710 JSR PC, (0) ;CALL THE MONITOR
006446 000207 18: RTS PC ;RETURN

```

```

006450 012777 00644 000356 POWDOWN: MOV #ILLUP, 2UPVEC ;SET FOR FAST UP
006456 012777 000340 000352 MOV #340, 2UPVEC+2 ;PRIO:7
006464 170246 STEPS -(6) ;GET THE FPS
006466 170011 SETD ;
006470 174046 STD ACO, -(6) ;SAVE AC'S
006472 174146 STD AC1, -(6)
006474 174246 STD AC2, -(6)
006476 174346 STD AC3, -(6)
006500 172404 LDD AC4, ACO
006502 174046 STD ACO, -(6)
006504 172405 LDD AC5, ACO
006506 174046 STD ACO, -(6)
006510 010046 MOV R0, -(6) ;SAVE REGISTERS
006512 010146 MOV R1, -(6)
006514 010246 MOV R2, -(6)
006516 010346 MOV R3, -(6)
006520 010446 MOV R4, -(6)
006524 010546 MOV R5, -(6)
006530 010667 000264 MOV SP, SAVE6 ;SAVE SP
006536 012777 006540 000276 MOV #POWUP, 2UPVEC ;SET UP VECTOR
006538 000000 HALT

006540 016706 000250 POWUP: MOV SAVE6, SP ;GET SP
006544 005001 CLR R1 ;WAIT LOOP FOR THE TTY
006546 005201 IS: INC R1
006550 001376 BNE IS
006552 012605 MOV (6)+, R5 ;GET THE REGISTERS
006554 012604 MOV (6)+, R4
006556 012603 MOV (6)+, R3
006560 012602 MOV (6)+, R2
006562 012601 MOV (6)+, R1
006564 012600 MOV (6)+, R0
006566 170011 SETD
006570 172426 LDD (6)+, ACO ;RESTORE THE AC'S
006572 174005 STD ACO, AC5
006574 172426 LDD (6)+, ACO
006576 174004 STD ACO, AC4
006600 172726 LDD (6)+, AC3
006602 172626 LDD (6)+, AC2
006604 172526 LDD (6)+, AC1
006606 172426 LDD (6)+, ACO
006610 170126 LDFPS (6)+ ;RESTORE FPS
006612 012777 006450 000210 MOV #POWDOWN, 2DOWNVEC ;SET UP THE POWER DOWN VECTOR
006620 012777 000340 000204 MOV #340, 2DOWNVEC+2
006624 000004 TYPE, .+2 ;.ASCIZ <15><12>"POWER"
006642 000002 RTI

006644 000000 ILLUP: HALT ;THE POWER UP SEQUENCE WAS STARTED
006646 000776 BR .-2 ;BEFORE THE POWER DOWN WAS COMPLETE

```

```

006650 010546 .IOT: MOV TTY, -(6) ;SAVE TTY
006652 017605 000002 MOV 22(6), TTY ;GET ADDRESS TO BE TYPED
006654 105715 1S: TSTB (TTY) ;TERMINATOR?
006656 001406 BEQ 2S
006658 112537 177566 MOVB (TTY)+, 2#177566 ;LOAD AND TYPE THE CHARACTER
006660 105737 177564 TSTB 2#177564 ;IS THE PRINTER READY
006662 100375 BPL .-4
006664 000770 BR 1S ;GET THE NEXT CHARACTER
006666 017646 000002 2S: MOV 22(6), -(6) ;GET ADDRESS TO BE TYPED
006668 062766 000002 000004 ADD 2, 4(6) ;ADD 2 TO THE ADDRESS
006670 022666 000002 CHP (6)+, 2(6) ;IS IT .+2?
006672 001076 BNE 3S ;NO
006674 000002 ADD 2, TTY ;ADD 2 TO THE ADDRESS
006676 000001 BIC 2, TTY ;BACK UP TO AN EVEN BYTE
006678 010566 000002 MOV TTY, 2(6) ;RESTORE ADDRESS
006680 010075 3S: MOV (6)+, TTY ;RESTORE TTY
006682 000002 RTI ;RETURN

006736 005015 020040 000040 SPACE: .ASCIZ <15><12>" "
006744 005015 000 RET: .ASCIZ <15><12>" "
006747 040 020052 000 $SIGN: .ASCIZ " #"
006753 015 020012 043040 FORTAN: .ASCIZ <15><12>" FORTAN: "
006760 051117 051124 047101
006766 003072 007040
006772 003440 000015 020040 FPUAN: .ASCIZ " = "<15><12>" FPU: "
007000 050106 035125 020040
007006 020040 020040 000

007014 000000 .EVEN
007016 177564 SAVE6: 0
007020 177566 TPS: 177564 ; TELEPRINTER STATUS REGISTER
007022 172160 TPB: 177566 ; TELEPRINTER DATA BUFFER
007024 000244 000246 FPTADR: 172160 ; FLOATING POINT ADDRESS ON THE 11/20
007030 000024 000026 FPVECT: 244, 246 ; FLOATING POINT VECTOR ADDRESS
007034 000024 000026 DMNVEC: 24, 26 ; POWER DOWN VECTOR ADDRESS
007040 000000 UPVEC: 24, 26 ; POWER UP VECTOR ADDRESS
007042 000000 .TYPE: 0
007044 000000 TRPB: 0
007046 000000 LAD: 0 ; LOOP ADDRESS
007050 000000 ERRORS: 0 ; ERROR COUNT
007052 000377 WC TS: 0 ; CONTAINS TYPEOUT INFO
000001 T! S: 377 ; ITERATION COUNT
.END

```


L03

FPVECT	007024	456#	457#	1553#										
HERE	004160	1024	1029	1035#										
HINUM	001012	414#	48#	558	633	690	749	827	909	971	1073	1074	1075	1076
		1220	1261											
HLT =	104000	391#	500	506	511	516	530	573	579	584	589	605	646	662
		703	721	764	770	775	780	799	842	848	853	858	881	922
		943	954	1009	1204	1209								
ICNT	001000	408#	458#	1040	1048	1050	1052#	1053#	1055	1058#	1059	1100		
ILLUP	006644	1470	1517#											
KIT	004300	1045	1051	1058#										
LAD	007044	459#	488#	561#	636#	693#	752#	830#	912#	974#	1054#	1060	1062	1558#
LONUM	001002	409#	484	557	632	689	748	826	908	970	1067	1068	1069	1070
		1219	1244	1260	1285	1311								
M1120	001104	437	442#											
MHEAD	005676	1301	1331	1336	1343#									
NOHEAD	005676	1330#												
NOP =	000040	389#												
OVER	004304	1341	1059#											
PC =	000007	370#	474#	547#	622#	679#	738#	816#	898#	960#	1031#	1454#	1467#	1468#
PODOWN	006450	448	1470#	1512										
POWUP	006540	1489	1492#											
PRINTR	006246	1309	1319	1324	1327	1334	1339	1342	1358	1361	1366	1369	1425#	
PRINTS	006256	1427#												
PS =	177776	386#	1160											
RANDOM2	005110	474	622	738	898	1213#								
RANDOM4	005246	547	679	816	960	1254#								
REPET2	005140	1223#	1245											
REPET4	005276	1264#	1286											
RET	006744	1303	1304	1540#										
RETAD	004526	1103	1111#											
PET1	001342	493#	1111											
RET10	003706	979#												
RET2	001630	566#	1112											
RET3	002126	641#												
RET4	002350	698#												
RET5	002602	757#	1115											
RET6	003112	835#	1116											
RET7	003442	917#												
RD =	000000	362#	451#	482#	555#	630#	687#	746#	824#	906#	968#	1028#	1120	1122
		1124	1126	1464#	1482	1501#								
R1 =	000001	363#	1100#	1101#	1102#	1483	1493#	1494#	1500#					
R2 =	000002	364#	1484	1499#										
R3 =	000003	365#	1042#	1485	1498#									
R4 =	000004	366#	475#	548#	623#	680#	739#	817#	899#	961#	1065	1071	1077	1081#
		1087#	1098	1108	1109#	1132#	1144#	1157	1163	1165#	1175#	1191	1194	1429
		1430#	1448	1453#	1486	1497#								
RS =	000005	367#	1487	1496#										
SAVE6	007014	1488#	1492	1549#										
SAVLAD	004260	1047	1053#	1061										
SCOPE =	104400	390#	467	532	608	665	724	802	884	946	1012			
SHIFT2	005146	1225#	1229											
SHIFT4	005304	1266#	1270											
SP =	000006	369#	436#	446#	1067#	1068#	1069#	1070#	1073#	1074#	1075#	1076#	1079	1080
		1083	1084	1085	1086	1104	1107	1156#	1159#	1172#	1190#	1193#	1488	1492#
SPACE	006736	1310	1317	1322	1325	1332	1337	1340	1359	1374	1367	1387	1391	1503#
SWR =	177570	385#	387	1014	1019	1038	1040	1042	1044	1046	1295	1300	1343	1348

MAINDEC-11-DCFPT-D FLOATING POINT MULTIPLY EXERCISER
DCFPT.E.P11 CROSS REFERENCE TABLE -- USER SYMBOLS

SOVRI	005026	1184	1187#											
SPOLSH	004326	475	548	623	680	739	817	899	961	1065#	1132	1144	1165	1175
SPOP2X	004374	480	628	741	904	1079#	1139	1172						
SPOP4X	004406	553	685	822	966	1083#	1151	1182						
SP.2A	004340	476	624	740	900	1069#	1133	1166						
SP.2B	004362	477	625	741	901	1075#	1134	1167						
SP.4A	004330	549	681	818	962	1067#	1145	1176						
SP.4B	004352	550	682	819	963	1073#	1146	1177						
SSB200	005036	1168	1178	1190#										
SSIGN	006747	1313	1541#											
STER	004472	1090	1100#											
STOV	004510	1091	1104#											
STOVI	004522	1106	1108#											
STST	004430	479	552	627	684	743	821	903	965	1089#	1138	1150	1171	1181
STYPE	005710	1312	1314	1316	1329	1348#								
SUNDRA	004666	1140	1152#											
SUNDR0	004600	1121	1130#											
SUNDR1	004634	1123	1142#											
SUNKER	004576	1128#												
S200A0	005044	1170	1180	1193#										
S200SB	004712	1137	1149	1159#										
S201SB	004732	1161	1163#											
.	= 007054	395#	396	397#	399#	403#	407#	488	505	510	561	578	583	636
		693	752	769	774	830	847	852	912	974	1093	1096	1203	1344
		1385	1426	1441	1444	1449	1456#	1514	1518	1526	1548#			
.EMT	004166	454	1038#											
.ET	005426	1296	1299#											
.IOT	006650	450	1520#											
.PR	006404	1425#	1427#	1428#	1442#	1443	1446#	1457#						
.STAT	005606	1321	1328#											
.TRP	005404	452	1295#											
.TYPE	007040	1016#	1017	1297#	1298	1404#	1412#	1417#	1420#	1421	1556#			

MAINDEC-11-DCFPT-D
 DCFPTE.P11

FLOATING POINT MULTIPLY EXERCISER
 CROSS REFERENCE TABLE -- PERMANENT SYMBOLS

AOC	789	867	868	869	933	995	996	997	1231	1233	1235	1236	1238	1241	1272
ADD	1274	1276	1277	1279	1282										
ASL	788	856	932	994	1156	1162	1193	1230	1232	1234	1237	1239	1240	1271	1273
BEQ	1275	1278	1280	1281	1529	1532									
BIC	1102	1225	1266	510	521	528	571	578	583	588	594	603	644	653	660
BIS	498	505	510	515	521	528	571	578	583	588	594	603	644	653	660
BISB	701	710	719	762	769	774	779	785	792	797	840	847	852	857	863
BIT	872	879	920	929	936	941	982	991	1000	1007	1029	1039	1041	1049	1061
BITB	1121	1123	1125	1127	1131	1143	1161	1184	1203	1296	1331	1441	1444	1461	1465
BMI	1523														
BNE	1533														
BPL	1023	1094	1097	1152	1185	1187	1420								
BR	1445														
BRB	649	706	925	987	1014	1019	1038	1044	1046	1089	1095	1130	1142	1160	1183
BRL	1202	1295	1300	1348											
BRS	1460														
BRSB	495	568	759	837	1090	1208	1351	1402							
BRSB	650	707	926	988	1015	1020	1045	1047	1051	1091	1096	1229	1245	1270	1286
BRL	1301	1349	1385	1395	1419	1447	1449	1463	1495	1531					
BR	1022	1093	1106	1321	1336	1344	1363	1526							
BRB	440	501	507	512	517	574	580	585	590	647	704	765	771	776	781
BRB	843	849	854	859	923	985	1140	1173	1205	1377	1405	1408	1410	1426	1432
CFCC	1518	1527													
CLR	503	520	527	576	593	602	652	659	709	718	767	784	791	796	845
CLR	862	871	878	928	935	940	990	999	1006						
CLR	458	459	469	470	471	472	473	532	542	543	544	545	546	607	617
CLR	618	619	620	621	664	674	675	676	677	678	723	733	734	735	736
CLR	737	801	811	812	813	614	815	883	893	894	895	896	897	945	955
CLR	956	957	958	959	1011	1018	1224	1265	1380	1381	1412	1427	1466	1493	
CLRB	1431	1433	1451												
CMF	497	504	509	514	570	577	582	587	643	700	761	768	773	778	839
CMF	846	851	856	919	981	1104	1107	1120	1122	1124	1126	1244	1285	1353	1396
CMFB	1397	1448	1462	1530											
CMFB	1040	1050													
CMFB	592	601	708	717	861	870	877	889	998	1005					
CMFB	519	526	651	658	783	790	795	827	934	939					
COM	1021	1386	1406												
DEC	1101	1394	1418												
EMT	391														
HALT	388	396	1128	1345	1490	1517									
INC	1228	1269	1299	1459	1494										
INCB	1053	1058	1442	1446											
IOT	392														
JMP	405	1035	1065	1071	1077	1098	1157	1163	1191	1194					
JSR	474	475	547	548	622	623	679	680	738	739	816	817	898	899	960
JSR	961	1031	1132	1144	1165	1175	1309	1312	1314	1316	1319	1324	1327	1329	1334
JSR	1339	1342	1352	1358	1361	1366	1369	1383	1388	1389	1390	1392	1393	1467	
LDD	557	558	559	565	689	690	691	697	826	827	828	834	970	971	972
LDD	978	1478	1480	1503	1505	1507	1508	1509	1510						
LDF	484	485	486	492	632	633	634	640	748	749	750	756	908	909	910
LDFPS	916														
LDFPS	483	487	556	560	631	635	688	692	747	751	825	829	907	911	969
LDFPS	973	1511													
LDUB	1043														
MOV	436	437	439	442	445	446	447	448	449	450	451	452	453	454	455
MOV	456	457	468	482	488	541	555	561	616	630	636	673	687	693	732

	746	752	810	824	830	892	906	912	954	968	974	1016	1024	1026	1028
	1054	1055	1059	1062	1067	1068	1069	1070	1073	1074	1075	1076	1079	1080	1083
	1084	1035	1086	1103	1153	1186	1213	1214	1215	1216	1217	1218	1219	1220	1221
	1222	1223	1242	1243	1246	1247	1248	1249	1250	1251	1254	1255	1256	1257	1258
	1259	1260	1261	1262	1263	1264	1283	1284	1287	1288	1289	1290	1291	1292	1297
	1305	1308	1311	1318	1323	1326	1333	1338	1341	1357	1360	1365	1368	1372	1373
	1374	1375	1376	1378	1379	1382	1403	1404	1407	1409	1411	1429	1430	1453	1464
	1470	1471	1482	1483	1484	1485	1486	1487	1488	1489	1492	1496	1497	1498	1499
	1500	1501	1512	1513	1520	1521	1528	1534	1535						
MOV8	1042	1052	1100	1307	1425	1428	1450	1524							
MULD	566	698	835	979											
MULF	493	641	757	917											
NOP	1032	1033	1034												
RESET	1030														
ROL	1226	1227	1267	1268	1413	1414	1415	1416	1417	1434	1436	1438			
ROLB	1435	1437	1439												
RTI	1025	1027	1036	1056	1063	1211	1346	1515	1536						
RTS	1081	1087	1109	1154	1188	1252	1293	1354	1370	1399	1423	1454	1468		
SBC	525	598	599	600	657	714	715	716	794	874	875	876	938	1002	1003
SETD	1004														
STD	1473	1502													
	572	596	604	702	712	720	841	865	880	983	993	1008	1474	1475	1476
	1477	1479	1481	1504	1506										
STF	499	523	529	645	655	661	763	787	798	921	931	942	1302		
STFPS	496	569	642	699	760	838	918	980	1200	1472					
STST	1201														
SUB	524	597	656	713	793	873	937	1001	1159	1190	1306				
TRAP	390														
TST	438	494	567	758	836	1060	1092	1108	1207	1343	1398	1401	1422		
TSTB	1048	1105	1320	1330	1335	1350	1362	1384	1440	1443	1522	1525			
.ASCIZ	1515	1539	1540	1541	1542	1545									
.RSECT	313														
.BLKW	1456														
.END	1562														
.EVEN	1515	1548													
.GLOBL	314														
.LIST	311	361	396	406	460	1013	1064	1212	1294	1347	1424	1469	1515	1519	
.MACR	361														
.MACRO	361														
.MLIST	311	361	396	406	460	1013	1064	1212	1294	1347	1424	1469	1515	1519	
.PAGE	725	831	1013												
.REPT	2	396													
.SBTTL	311	361	406	460	1013	1064	1212	1294	1347	1424	1469	1519			
.TITLE	312														

. ABS. 007054 000
000000 001

ERRORS DETECTED: 0
DEFAULT GLOBALS GENERATED: 0

* DCFPTE.SEQ/SOL/CRF/PAGNUM=DCFPTE
RUN-TIME: 6 11 2 SECONDS
RUN-TIME RATIO: 384/20=18.9

MAINDEC-11-DCFPT-D FLOATING POINT MULTIPLY EXERCISER
DCFPT.E.P11 CROSS REFERENCE TABLE -- PERMANENT SYMBOLS

CORE USED: 9K (17 PAGES)

E04
MACY11 27(732) 17-SEP-76 10:19 PAGE 46

