

# KD11-K

PDP11/6X FP11E FP BASIC  
MD-11-DQFPA-A

EP-DQFPA-A-DL-A

APR 1977

COPYRIGHT © 1977



FICHE 1 OF 1

MADE IN US

This image shows a microfilm grid with 12 columns and 16 rows of frames. Each frame contains a different view of data, including tables, diagrams, and text. The data is presented in a structured, grid-like format, typical of microfilm documentation. The frames are arranged in a regular pattern, with each frame containing a distinct set of information. The overall appearance is that of a technical or scientific data archive.

B01

EOF1DFK7BR580411  
DQFPA.MEM

09-FEB-77 09:46

00010000C INSTR70883S

MACY@P20(9006) 09-FEB-77 09:46 PAGE 1

00010000

770323

000000

.REPT 0

IDENTIFICATION

PRODUCT CODE:	MAINDEC-11-DQFPA-A-D
PRODUCT NAME:	PDP-11/6X - FP11-E FLOATING POINT UNIT BASIC INSTRUCTION TESTS
DATE :	MARCH 1977
MAINTAINER:	DIAGNOSTIC GROUP
AUTHOR:	DONALD NORTH

COPYRIGHT (C) 1977  
DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS

THIS SOFTWARE IS FURNISHED TO THE PURCHASER UNDER A LICENSE FOR USE ON A SINGLE COMPUTER SYSTEM, AND CAN BE COPIED (WITH INCLUSION OF DIGITAL'S COPYRIGHT NOTICE) ONLY FOR USE IN SUCH SYSTEM, EXCEPT AS MAY OTHERWISE BE PROVIDED IN WRITING BY DIGITAL.

THE INFORMATION IN THIS DOCUMENT IS SUBJECT TO CHANGE WITHOUT NOTICE, AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT CORPORATION. DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY FOR ANY ERRORS THAT MAY APPEAR IN THIS DOCUMENT.

DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS SOFTWARE ON EQUIPMENT NOT SUPPLIED BY DIGITAL.

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46

47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60  
61  
62  
63  
64  
65  
66  
67  
68  
69  
70  
71  
72  
73  
74  
75  
76  
77  
78  
79  
80  
81  
82  
83  
84  
85  
86  
87  
88  
89  
90  
91  
92  
93  
94  
95  
96  
97  
98  
99  
100

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/OPERATOR ACTION
- 5. OPERATING PROCEDURE
  - 5.1 OPERATIONAL SWITCH SETTINGS
  - 5.2 PROGRAM/OPERATOR ACTION
  - 5.3 HOT (FP11-E) / WARM (PDP-11/6X) SELECTION
- 6. ERRORS
  - 6.1.1 ERROR MESSAGE FORMAT
  - 6.1.2 FLOATING POINT DATA FORMAT
  - 6.2 RECOVERY
  - 6.3 CAUSES
- 7. RESTRICTIONS
  - 7.1 STARTING
  - 7.2 OPERATIONAL
- 8. MISCELLANEOUS
  - 8.1 EXECUTION TIME
  - 8.2 STACK POINTER
  - 8.3 POWER FAIL
- 9. PROGRAM DESCRIPTION
  - 9.1 ORGANIZATION
  - 9.2 TEST DESCRIPTION
  - 9.3 SUBROUTINE ABSTRACTS
- 10. ACT/APT/XXDP

101  
102  
103  
104  
105  
106  
107  
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  
141  
142  
143  
144  
145  
146  
147  
148  
149  
150  
151  
152  
153  
154  
155

## 1. ABSTRACT

THIS PROGRAM IS THE BASIC FUNCTIONAL TEST FOR THE PDP-11/6X FLOATING POINT PROCESSOR. FUNCTIONALITY TESTS OF ALL STATUS REGISTERS AND ACCUMULATORS ARE PERFORMED TO VERIFY THEIR OPERATION (EG, RIPPLING BIT TESTS, ALTERNATING BIT TESTS, UNIQUE REFERENCE TESTS). ALL ADDRESS MODES (SOURCE, DESTINATION, FLOATING SOURCE, FLOATING DESTINATION) ARE TESTED FOR CORRECT OPERAND REFERENCE, AND VERIFICATION OF SIDE AFFECTS. FINALLY, THE BASIC NO-OPERAND AND SINGLE OPERAND INSTRUCTIONS ARE TESTED TO INSURE THEIR FULL FUNCTIONALITY IN ALL PDP-11/6X FPU MODES. BOTH "HOT" (FP11-E OPTION) AND "WARM" (PDP-11/6X MICROCODE) FLOATING POINT UNITS CAN BE SELECTED FOR TESTING.

## 2. REQUIREMENTS

## 2.1 EQUIPMENT

PDP-11/6X STANDARD COMPUTER WITH MINIMUM 16K OF MEMORY. OPTIONAL FP11-E FLOATING POINT UNIT, IF SELECTED.

## 2.2 STORAGE

THE PROGRAM USES MEMORY 0-33566(8). THE UPPER 2.0K WORDS ARE RESERVED FOR THE XXDP MONITOR, IF EMPLOYED.

## 2.3 PRELIMINARY PROGRAMS

THE CPU, CACHE, AND MEMORY TEST PROGRAMS MUST BE RUN FIRST TO VERIFY THE CORRECT OPERATION OF THE BASE MACHINE.

THE PDP-11/6X - FP11-E FLOATING POINT PROCESSOR INSTRUCTION SET TESTS SHOULD THEN BE RUN IN THE FOLLOWING ORDER:

- (1) DQFPA FPU BASIC INSTRUCTION TESTS
- (2) DQFPB FPU ADVANCED INSTRUCTION TESTS
- (3) DQFPC FPU INSTRUCTION EXERCISER
- (4) DQFPD FPU ADD/SUB/MUL/DIV RANDOM EXERCISER

## 3. LOADING PROCEDURE

USE THE STANDARD PROCEDURE FOR ABSOLUTE TAPES, OR LOAD VIA XXDP MEDIA.

## 4. STARTING PROCEDURE

MAINDEC-11-DQFPA-A

PAGE 4

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  
199  
200  
201  
202  
203  
204  
205  
206  
207  
208  
209  
210

## 4.1 CONTROL SWITCH SETTINGS

SEE SECTION 5.1  
SWITCH REGISTER (000000) IS WORST CASE TEST.

## 4.2 STARTING ADDRESS

THE PROGRAM MUST ALWAYS BE STARTED AT LOCATION 200(8).

## 4.3 PROGRAM/OPERATOR ACTION

LOADING VIA ABSOLUTE PAPERTAPE:

- (1) LOAD PROGRAM INTO MEMORY USING ABS LOADER.
- (2) LOAD ADDRESS 200 (8).
- (3) SET SWITCHES (SEE SECTION 5.1)  
SR=(000000) IS WORST CASE TEST.
- (4) PRESS CONTROL/START TO BEGIN.
- (5) PROGRAM TYPES IDENTIFICATION HEADER (VERIFY THAT THE  
CORRECT PROGRAM HAS BEEN LOADED!), AND EXECUTION BEGINS.

## 5. OPERATING PROCEDURE

## 5.1 OPERATIONAL SWITCH SETTINGS

THE DEFINITION OF THE SPECIFIC BITS IN THE SWITCH REGISTER  
(EITHER HARDWARE OR SOFTWARE) ARE AS FOLLOWS:

SW15=1	100000	MALT ON ERROR
SW14=1	040000	LOOP ON CURRENTLY EXECUTING TEST
SW13=1	020000	INHIBIT ERROR TYPEOUTS (WHICH IS AN "ERROR MESSAGE" RESULTING FROM AN ERROR DETECTED IN THE HARDWARE)
SW12=1	010000	INHIBIT STATUS TYPEOUTS (WHICH IS A NON-ERROR RELATED INFORMATIVE MESSAGE, SUCH AS "END PASS #XXX")
SW11=1	004000	INHIBIT ITERATIONS PER TEST
SW10	002000	SET=BELL ON ERROR/CLEAR=BELL ON PASS END
SW09=1	001000	LOOP ON ERROR
SW08=1	000400	LOOP ON TEST NUMBER IN "SLPTST" IF SET, THEN THE TEST SPECIFIED BY THE TEST NUMBER CONTAINED IN THE MEMORY WORD "SLPTST" (SEE PROGRAM LISTING) WILL SPECIFY THE DESIRED TEST ON WHICH TO LOOP.
SW01	000002	CLEAR=TEST HOT-FP/WARM-FP ALTERNATELY EACH PASS (IE, PASS#1 HFP, PASS#1 WFP, PASS#2 HFP, PASS#2 WFP, ETC)
SW00	000001	SET=TEST ONLY UNIT SPECIFIED IN SW00 SET=SELECT WARM FP, IF SW01=1 CLEAR=SELECT HOT FP, IF SW01=1

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

\*NOTE\* FOR SW01, SW00 - IF NO HOT FP (FP11-E) IS PRESENT, THEN WARM FP (PDP-11/6X MICROCODE) IS AUTOMATICALLY SELECTED.

## 5.2 PROGRAM/OPERATOR ACTION

ONCE EXECUTION HAS BEGUN, MINIMAL OPERATOR INTERVENTION IS REQUIRED, UNLESS THE PROGRAM DETECTS AN ERROR IN THE HARDWARE.

IF ALL IS WELL, THE PROGRAM TYPES ITS NAME UPON BEGINNING; AND AT THE START OF EACH PASS, THE CURRENT PASS NUMBER (IN OCTAL) IS ECHOED. NOTE THAT SETTING SW<12>=1 WILL INHIBIT THE TYPEOUT OF THE BEGIN AND END PASS MESSAGES.

IF SW<10>=0, THE CONSOLE BELL WILL BE RUNG AT THE END OF EACH PASS. NOTE THAT ONLY SW<10> AFFECTS THE BELL RINGING AT END OF PASS - SW<12> HAS NO EFFECT ON THIS FUNCTION.

IF AN ERROR OCCURS DURING EXECUTION, MANY VARIATIONS IN ACTION ARE POSSIBLE DEPENDING UPON THE SWITCH SETTINGS.

SW<15>=1 WILL CAUSE THE CPU TO HALT AFTER AN ERROR.

SW<13>=1 WILL ALSO INHIBIT ANY ERROR MESSAGE TYPEOUT THAT WOULD OCCUR AT THIS TIME.

SW<10>=1 WILL CAUSE THE CONSOLE BELL TO BE RUNG ONLY WHEN AN ERROR IS DETECTED (AND NOT AT THE END OF A PASS).

SW<9>=1 CAUSES THE PROGRAM TO LOOP ON THE MOST RECENT ERROR, AS LONG AS IT CONTINUES TO OCCUR.

THERE ARE ALSO SEVERAL OTHER GENERAL USE FUNCTIONS DEFINED BY THE SWITCHES:

SW<11>=1 WILL INHIBIT THE ITERATIONS (=2000(10)) PERFORMED OF EACH TEST ON PASSES 2, 3, 4 THRU THE PROGRAM.

SW<14>=1 CAUSES THE PROGRAM TO LOOP INDEFINATELY ON THE CURRENTLY EXECUTING TEST.

SW<8>=1 CAUSES THE PROGRAM TO CONTINUE EXECUTION AS NORMAL, EXCEPT WHEN THE CONTENTS OF MEMORY WORD "SLPTST" MATCHES THE NUMBER OF THE TEST CURRENTLY EXECUTING. AT THIS POINT, THE TEST IS LOOPED ON INDEFINATELY, UNTIL EITHER SW<8>=0 OR "SLPTST" IS CHANGED. NOTE THAT IF "SLPTST" DOES NOT MATCH THE TEST NUMBER OF ANY TEST, THE CONTENTS OF "SLPTST" ARE EFFECTIVELY IGNORED, AND EXECUTION PROCEEDS NORMALLY.

## 5.3 HOT (FP11-E) / WARM (PDP-11/6X) SELECTION

WHEN THE PROGRAM IS STARTED (AT 200(8)), A MESSAGE IS OPTIONALLY PRINTED INDICATING THE PRESCENCE/ABSCENCE OF AN FP11-E HOT FLOATING POINT UNIT OPTION (BASED UPON WHETHER "WHAMI" BIT<04> IS 1/0 RESPECTIVELY).

IF NO FP11-E HOT FP OPTION IS PRESENT, THE MESSAGE IS TYPED, AND ANY ATTEMPTS TO SELECT IT FOR TESTING VIA SW01 AND SW00 ARE IGNORED. ONLY WARM FP (PDP-11/6X MICROCODE) FLOATING POINT CAN BE TESTED/SELECTED.

267  
268  
269  
270  
271  
272  
273  
274  
275  
276  
277  
278  
279  
280  
281  
282  
283  
284  
285  
286  
287  
288  
289  
289  
290  
291  
292  
293  
294  
295  
296  
297  
298  
299  
300  
301  
302  
303  
304  
305  
306  
307  
308

IF THE FP11-E IS PRESENT, TEST SELECTION IS AS FOLLOWS:

WHEN SW01=0, THE HOT AND WARM FLOATING POINT UNITS ARE TESTED ALTERNATELY EACH PASS - IN THE ORDER (1) HOT, THEN (2) WARM. NOTE THAT EACH "PASS" NOW CONSISTS OF TWO SEPARATE SUB-PASSES.

WHEN SW01=1, THEN DEDICATED SELECTION OF A PARTICULAR UNIT IS SPECIFIED IN SW00:

SW00=0 --> TEST HFP FP11-E OPTION ONLY  
SW00=1 --> TEST WFP PDP-11/6X MICROCODE ONLY

## 6. ERRORS

### 6.1 FORMAT OF MESSAGES

#### 6.1.1 ALL ERROR MESSAGES CONSIST OF THREE LINES OF DATA:

THE FIRST LINE IS A BRIEF MESSAGE WHICH EXPLAINS WHAT ERROR WAS DETECTED (EG, THE RESULT OF THE "ABSF" INSTRUCTION WAS BAD).

THE PREFIX "HOT:" OR "WARM:" IS ALSO ATTACHED TO THE MESSAGE TO INDICATE THE SOURCE OF THE ERROR; THE FP11-E UNIT OR THE PDP-11/6X RESPECTIVELY.

THE SECOND LINE CONSISTS OF DATA HEADERS TO IDENTIFY THE VALUES TYPED OUT ON LINE THREE. THESE HEADERS WILL EITHER BE OF THE FORM "EXPECTED" AND "RECEIVED" DATA, OR WILL BE A MNEMONIC NAME OF A WORD LOCATION IN MEMORY OR REGISTERS.

THE THIRD LINE DISPLAYS THE CONTENTS OF THE LOCATIONS SPECIFIED BY LINE TWO AS SIX DIGIT OCTAL NUMBERS. NOTE THAT ALL DATA DISPLAYED IN ANY MESSAGES ARE OCTAL NUMBERS.

AS EXPLAINED IN SECTION 5.2, SETTING SW<13>=1 WILL SUPPRESS THE TYPING OF THESE MESSAGES.

309  
310  
311  
312  
313  
314  
315  
316  
317  
318  
319  
320  
321  
322  
323  
324  
325  
326  
327  
328  
329  
330  
331  
332  
333  
334  
335  
336  
337  
338  
339  
340  
341  
342  
343  
344  
345  
346  
347  
348  
349  
350  
351  
352  
353  
354  
355  
356  
357  
358  
359  
360  
361  
362  
363

6.1.2 FLOATING POINT UNIT DATA FORMATS:

FLOATING POINT STATUS WORD (FPS):

BIT##	OCTAL	FUNCTION
15	100000	FER - FLOATING ERROR FLAG SET WHEN EITHER FIUV, FIU, FIV, FIC ENABLED AND APPROPRIATE EXCEPTION OCCURRED.
14	040000	FID - FLOATING DISABLE INTERRUPTS NO FP INTERRUPTS TO VECTOR 244(8) IF SET.
13, 12		NOT USED
11	004000	FIUV - FLOATING UNDEFINED VARIABLE INTERRUPT IF SET, (-0) MEMORY DATA IS ERROR
10	002000	FIU - FLOATING INTR UNDERFLOW IF SET AND UNDERFLOW, SET FER, STORE ANSWER, EXPONENT WRONG BY +400(8) IF CLEAR AND UNDERFLOW, ANSWER <-- ZERO
9	001000	FIV - FLOATING OVERFLOW INTERRUPT IF SET AND OVERFLOW, SET FER, STORE ANSWER, EXPONENT WRONG BY -400(8) IF CLEAR AND OVERFLOW, ANSWER <-- ZERO
8	000400	FIC - FLOATING INTEGER CONVERSION INTERRUPT IF SET AND "STCFI" ERROR, ANSWER <-- ZERO, SET ERROR IF CLEAR AND "STCFI" ERROR, ANSWER <-- ZERO
7	000200	FD - FLOATING MODE 1=DOUBLE, 64 BIT OPERANDS (4W) 0=SINGLE, 32 BIT OPERANDS (2W)
6	000100	FL - INTEGER MODE 1=LONG, 32 BIT INTEGERS (2W) 0=SHORT, 16 BIT INTEGERS (1W)
5	000040	FT - ROUND/TRUNCATE MODE 1=TRUNCATE RESULTS 0=ROUND RESULTS
4	000020	FMM - PUT FP11-E ONLY IN MAINTENANCE MODE
3:0	000017	FN-FZ-FV-FC - FLOATING CONDITION CODES

FLOATING EXCEPTION CODES (FEC):

OCTAL	ENABLE	(NOT USED)
00	(NONE)	(NOT USED)
02	(NONE)	FP OPCODE ERROR
04	(NONE)	FP DIVIDE-BY-ZERO ERROR
06	W/FIC	FP INTEGER CONVERSION ERROR
10	W/FIV	FP OVERFLOW ERROR
12	W/FIU	FP UNDERFLOW ERROR
14	W/FIUV	FP UNDEFINED-VARIABLE/(-0) ERROR
16	W/FMM	FP MAINTENANCE TRAP

NOTE - IN "FEC" CODE TYPEOUTS IN ERROR MESSAGES ONLY THE LOW ORDER BYTE IS USED - IGNORE THE PROGRAM FLAG BIT IN THE UPPER BYTE.



364  
365  
366  
367  
368  
369  
370  
371  
372  
373  
374  
375  
376  
377  
378  
379  
380  
381  
382  
383  
384  
385  
386  
387  
388  
389  
390  
391  
392  
393  
394  
395  
396  
397  
398  
399  
400  
401  
402  
403  
404  
405  
406  
407  
408  
409  
410  
411  
412  
413  
414  
415  
416  
417  
418

## FLOATING POINT DATA:

IN FLOAT MODE (FD=0), IS 2-16. BIT WORDS, 32. BITS  
IN DOUBLE MODE (FD=1), IS 4-16. BIT WORDS, 64. BITS

## FIRST WORD: (BOTH F, D MODES)

B15=SIGN OF NUMBER (1/-, 0/+)  
B14:07=EXPONENT, 8.BITS, FROM -128./+127.  
B06:00=FRACTION, 7.BITS

## SECOND WORD: (BOTH F, D MODES)

B15:00=FRACTION, 16.BITS

## THIRD, FOURTH WORDS: (ONLY D MODE)

B15:00, B15:00=FRACTION, 32. BITS

IN F MODE, THE COMPOSITE 24. BIT FRACTION  
IS FORMED BY:

.1#[WORD1-BIT<06:00>]#[WORD2-BIT<15:00>]

IN D MODE, THE COMPOSITE 56. BIT FRACTION  
IS FORMED BY:

.1#[WORD1-BIT<06:00>]#[WORD2-BIT<15:00>]  
#[WORD3-BIT<15:00>]#[WORD4-BIT<15:00>]

FOR A MORE DETAILED OPERATION/EXPLANATION OF FLOATING POINT  
DATA FORMATS AND OPERATIONS, SEE THE PDP-11/6X PROCESSOR  
HANDBOOK SECTION ON THE FLOATING POINT INSTRUCTION SET.

## 6.2 RECOVERY

RECOVERY FROM ERRORS HAS BEEN ATTEMPTED TO BE MADE AS  
AUTOMATIC AND EFFORTLESS AS POSSIBLE. HOWEVER, IN MANY CASES,  
DUE TO THE NATURE OF THE ERROR, THE PROGRAM MAY NOT EVEN BE  
ABLE TO BE RUN (EG, IF THE FLOATING POINT MODULE IS IN A HUNG  
STATE, AND CAN NEVER ENTER THE READY STATE TO ACCEPT A NEW FPP  
INSTRUCTION). AT THIS POINT, SOLVING THE PROBLEM IS A DIRECT  
FUNCTION OF THE OPERATORS' INGENUITY. THIS TEST SERIES HAS  
BEEN DESIGNED TO TEST THE FLOATING POINT PROCESSOR SO THAT  
THESE TYPES OF FAILURES TO RUN WILL BE MINIMAL. THE TESTS  
HAVE BEEN PLACED IN A SPECIFICALLY STRUCTURED SEQUENCE IN THE  
PROGRAM TO IMPLEMENT THIS STRATEGY: TESTING THE MOST BASIC  
ELEMENTS FIRST, PROCEEDING UPWARD IN COMPLEXITY AFTER  
ESTABLISHING THEIR CORRECT OPERATION. THIS IS WHY IT IS  
EXTREMELY IMPORTANT THAT THE FLOATING POINT TEST PROGRAMS BE  
(1) RUN IN THE PRESCRIBED ORDER, AND (2) ONLY BE STARTED AT  
THEIR BEGINNING ADDRESS (USUALLY 200(8)). THE PROGRAM WILL  
DISPLAY, AT AN ERROR, THE MOST PERTINENT INFORMATION RELATING  
TO THE ERROR, AND A BRIEF EXPLANATION OF THE FAILING FUNCTION.

## 6.3 CAUSES

419  
420  
421  
422  
423  
424  
425  
426  
427  
428  
429  
430  
431  
432  
433  
434  
435  
436  
437  
438  
439  
440  
441  
442  
443  
444  
445  
446  
447  
448  
449  
450  
451  
452  
453  
454  
455  
456  
457  
458  
459  
460  
461  
462  
463  
464  
465  
466  
467  
468  
469  
470  
471  
472  
473  
474

THESE TEST PROGRAMS ARE NOT HARDWARE ORIENTED, AND AS SUCH IT IS NOT POSSIBLE TO CALL OUT PARTICULAR HARDWARE AREAS AND MODULES RELATING TO A GIVEN FUNCTIONAL FAILURE. HARDWARE DIAGNOSIS FOR A PARTICULAR MACHINE MUST BE DONE USING THE APPROPRIATE ENGINEERING ROM FLOWS AND PRINTS, ALONG WITH THE KNOWN FUNCTIONAL ERRORS (AS DETECTED BY THE PROGRAMS). THIS IS THE INTENT UNDER WHICH THESE INSTRUCTION TESTS WERE DESIGNED AND CODED.

7. RESTRICTIONS

7.1 STARTING

THE PROGRAM MUST BE STARTED AT LOCATION 200(8) ALWAYS.

7.2 OPERATIONAL

THERE ARE NO OPERATIONAL RESTRICTIONS.

8. MISCELLANEOUS

8.1 EXECUTION TIME

-----  
AVERAGE EXECUTION TIME PER PASS

MODEL	SHORTEST PASS	LONGEST PASS
PDP-11/6X	1 SEC	1 MIN:30 SEC
PDP-11/6X W/FP11-E	1 SEC	X MIN:XX SEC

-----  
SEC = SECONDS / MIN = MINUTES

SHORTEST PASS ::= NO ITERATIONS, USING SWR=(004000)

LONGEST PASS ::= 2000(10) ITERATIONS/TEST, USING SWR=(000000)

8.2 STACK POINTER

THE STACK POINTER IS SET TO 1100(8) AT THE START OF EACH PASS. IF ALL IS OPERATING CORRECTLY, IT SHOULD ALSO BE THIS VALUE AT THE START OF EACH TEST, AND AT THE END OF A PASS.

8.3 POWER FAIL

THE TESTS MAY BE POWER FAILED AT ANY TIME. SPURIOUS ERROR

475  
476  
477  
478  
479  
480  
481  
482  
483  
484  
485  
486  
487  
488  
489  
490  
491  
492  
493  
494  
495  
496  
497  
498  
499  
500  
501  
502  
503  
504  
505  
506  
507  
508  
509  
510  
511  
512  
513  
514  
515  
516  
517  
518  
519  
520  
521  
522  
523  
524  
525  
526  
527  
528  
529  
530

MESSAGES MAY OCCUR IF THE FAILURE OCCURRED WHILE THE F.P.U. WAS EXECUTING A FUNCTION, AS NONE OF ITS REGISTERS (FPS, FEC, FEA, ACCUMULATORS) ARE SAVED IN THE EVENT OF A POWER FAILURE. HOWEVER, THESE MESSAGES SHOULD ONLY OCCUR ONCE (IF AT ALL) IMMEDIATELY AFTER POWER IS RESTORED. WHEN POWER IS RESTORED, "POWER" IS TYPED ON THE CONSOLE AND EXECUTION CONTINUES WHERE IT WAS INTERRUPTED.

NOTE THAT THE "VOLATILE" SWITCH REGISTER CONTENTS ARE SAVED AND RESTORED FROM THE STACK IN A POWER FAIL SEQUENCE. THEREFORE THE SWITCH REGISTER SETTINGS SHOULD NOT BE LOST OVER A POWER FAIL.

## 9. PROGRAM DESCRIPTION

### 9.1 ORGANIZATION

THESE PROGRAMS ARE ORGANIZED AS MUCH AS POSSIBLE IN A STRAIGHTFORWARD, LINEAR MANNER. THE MAIN BODY OF CODE IS STRUCTURED AS FOLLOWS:

- (1) INITIALIZATION ROUTINE
  - SETS UP VECTORS, TYPES HEADER, ETC.
- (2) MAIN BODY OF TESTS
  - INLINE TEST CODE, INLINE TEST CALLS
- (3) END OF PASS ROUTINE
  - END OF PASS PROCESSING
- (4) TEST SUBROUTINES
  - SUBROUTINES CONTAINING COMMON TEST CODE
- (5) OVERHEAD ROUTINES
  - SERVICE SUBROUTINES (TYPEOUT, ETC.)

WHEREVER FEASIBLE, COMMON SECTIONS OF CODE FOR WIDELY USED FUNCTIONS ARE CONDENSED INTO SUBROUTINES TO CONSERVE MEMORY. THIS INCLUDES NOT ONLY STANDARD SERVICE ROUTINES (SUCH AS SCOPE, ERROR, AND ASCII TYPEOUT), BUT ALSO TESTING ROUTINES WHICH PERFORM VERY SIMILAR FUNCTIONS. THUS IN MANY CASES (THE "ADDF" INSTRUCTION TESTING, FOR EXAMPLE) A SINGLE BODY OF CODE (A SUBROUTINE) IS USED TO PERFORM ALL THE FUNCTIONAL TESTS, WITH A VARIABLE PARAMETER LIST PASSED AT EACH CALL CONTAINING THE DATA OPERANDS AND EXPECTED RESULT FOR EACH INDIVIDUAL TEST. THIS CONSTRUCTION FACILITATES THE ADDITION/DELETION OF TESTS (SHOULD THAT EVER BE NECESSARY), AND ALSO GREATLY CONSERVES MEMORY SPACE REQUIREMENTS WHEN A LARGE NUMBER OF CALLS TO A GIVEN BODY OF CODE ARE REQUIRED.

THE INDIVIDUAL TESTS WITHIN EACH PROGRAM HAVE ALSO BEEN SEQUENCED IN A PARTICULAR ORDER TO FACILITATE THE DETECTION AND RESOLUTION OF ERRORS AS QUICKLY AS POSSIBLE. EACH OF THE TESTS BEGINS AS SIMPLY AS POSSIBLE, FIRST TESTING THE MOST

531  
532  
533  
534  
535  
536  
537  
538  
539  
540  
541  
542  
543  
544  
545  
546  
547  
548  
549  
550  
551  
552  
553  
554  
555  
556  
557  
558  
559  
560  
561  
562  
563  
564  
565  
566  
567  
568  
569  
570  
571  
572  
573  
574  
575  
576  
577  
578  
579  
580  
581  
582  
583  
584  
585  
586

BASIC ELEMENTS. MORE COMPLEX ELEMENTS ARE TESTED AFTERWARDS, EMPLOYING A PHILOSOPHY THAT THE SIMPLER THE TEST, THE BETTER THE RESOLUTION. ALL FUNCTIONS ARE EVENTUALLY TESTED, BUT HOPEFULLY MOST ERRORS WILL BE CAUGHT AND CORRECTED EARLY. A MUCH MORE DETAILED ANALYSIS OF THE SEQUENCE OF TESTS PERFORMED IS PRESENTED IN SECTION 9.2 .

## 9.2 TEST DESCRIPTION

THIS DIAGNOSTIC IS STRUCTURED TO SEQUENTIALLY PERFORM THE FOLLOWING SERIES OF TESTS:

- (1) FLOATING POINT STATUS REGISTER FUNCTIONALITY.  
ALTERNATING ONES/ZEROS DATA PATTERNS WITH 'LDFPS/STFPS' INSTRUCTIONS IN DEFINED BIT POSITIONS OF REGISTER. 'SET-' MODE CHANGE INSTRUCTIONS (F-D/I-L) TO FPS REGISTER.
- (2) 16. BIT OPERAND FETCH/STORE.  
USING 'LDFPS/STFPS' INSTRUCTION SEQUENCES, CHECK 16. BIT OPERAND FETCH AND STORE USING ALL PDP-11 ADDRESS MODES. VERIFY OPERAND REFERENCE, SIDE EFFECTS FOR AUTO INCREMENT AND DECREMENT OF REGISTERS.
- (3) 'CFCC' INSTRUCTION  
VERIFY FLOATING POINT CONDITION CODES WITH ALTERNATING ONES/ZEROS DATA PATTERNS; 'CFCC' INSTRUCTION FUNCTIONALITY OF FLOATING C.C. -> BASE MACHINE C.C.
- (4) FLOATING POINT ACCUMULATORS  
USING ALTERNATING ONES/ZEROS DATA PATTERNS AND 'LDD/STD' SEQUENCES VERIFY FLOATING POINT REGISTERS 64. BIT DATA AND 32. BIT DATA MODES.
- (5) 64. BIT OPERAND FETCH/STORE.  
USING 'LDD/STD' INSTRUCTION SEQUENCES, CHECK 64. BIT OPERAND FETCH AND STORE USING ALL PDP-11 ADDRESS MODES. VERIFY OPERAND REFERENCE, SIDE EFFECTS FOR AUTO INCREMENT AND DECREMENT OF REGISTERS (2/8 CONSTANTS).
- (6) 32. BIT OPERAND FETCH/STORE.  
USING 'LDF/STF' INSTRUCTION SEQUENCES, CHECK 32. BIT OPERAND FETCH AND STORE USING ALL PDP-11 ADDRESS MODES. VERIFY OPERAND REFERENCE, SIDE EFFECTS FOR AUTO INCREMENT AND DECREMENT OF REGISTERS (2/4 CONSTANTS).
- (7) FLOATING ACCUMULATOR ADDRESSING.  
VERIFY FP ACCUMULATOR ADDRESSING BY WRITING ACCUMULATOR (- ADDRESS(ACCUMULATOR), CHECK ACC-5.
- (8) SINGLE OPERAND FP INSTRUCTIONS.  
VERIFY FUNCTIONALITY OF 'ABS-' 'NEG-' 'TST-', AND 'CLR-' FP INSTRUCTIONS IN F/D

MODES. CHECK FLOATING CC SETTINGS AND  
EXCEPTION CONDITIONS, WHEN APPLICABLE (-0  
ONLY). TESTED VIA SUBROUTINES.

9.3 SUBROUTINE ABSTRACTS

9.3.1 TRAPCATCHER

THE TRAPCATCHER IS A SERIES OF INSTRUCTIONS OCCUPYING THE INTERRUPT VECTOR AREA OF MEMORY. IT CONSISTS OF THE SEQUENCE:

```
.WORD .+2 ;PC AFTER TRAP
.WORD 0 ;PS AFTER TRAP
```

PLACED AT EACH VECTOR ADDRESS IN LOCATIONS 4-776(8) OF MEMORY. THE FIRST WORD OF EACH PAIR ("PC AFTER TRAP") POINTS TO THE SECOND WORD, WHICH SERVES A DUAL PURPOSE AS (1) THE NEW LOADED PS (ALL ZEROS), AND (2) THE NEXT INSTRUCTION TO EXECUTE (0=HALT).

WHEN THE PROGRAM IS EXECUTING, ANY REQUIRED VECTORS ARE SET UP IN THE VECTOR AREA WITH APPROPRIATE VALUES; THE OTHERS BEING LEFT IN THE "TRAPCATCHER" STATE. THUS, IF AN UNEXPECTED TRAP EVER OCCURS IN THE MACHINE, IT WILL BE CAUGHT, AND THE MACHINE SUBSEQUENTLY HALTED, DISPLAYING THE VECTOR ADDRESS \* PLUS FOUR \* IN THE ADDRESS LIGHTS.

9.3.2 SCOPE ROUTINE - \$SCOPE

THE SCOPE ROUTINE IS ENTERED FROM THE FIRST INSTRUCTION OF EACH TEST IN THE PROGRAM. (NOTE THAT BY DEFINITION, A "TEST" WILL BE DESIGNATED AS THE SECTION OF CODE BETWEEN TWO "SCOPE" STATEMENTS.) THIS ROUTINE PROVIDES THE OVERHEAD CODE NECESSARY TO IMPLEMENT SEVERAL OF THE SWITCH REGISTER CONTROL OPTIONS. UPON ENTRANCE TO A TEST, THE SCOPE STATEMENT AT THE BEGINNING SETS UP CERTAIN LOCATIONS (SEE BELOW) TO SPECIFY THE CURRENT TEST NUMBER AND LOOPING ADDRESS (FOR ITERATIONS). CONTROL IS THEN PASSED TO THE ACTUAL TEST CODE, PERFORMING THE DESIRED TEST. UPON EXIT, THE SCOPE STATEMENT OF THE NEXT TEST IS ENTERED, WHICH DETERMINES WHETHER TO (1) LOOP BACK TO THE PREVIOUS TEST (EG, FOR ITERATIONS) OR (2) INITIALIZE FOR THE NEXT TEST (AS DESCRIBED EARLIER, ABOVE).

ENTRANCE TO THE SCOPE ROUTINE IS VIA AN "IOT" TRAP CALL THROUGH LOCATION 20(8). (FROM THE SCOPE=IOT EQUATE). DEPENDING UPON THE SWITCH SETTINGS (SEE 5.2), CODE IS PRESENT TO: LOAD THE FPU MICRO BREAK REGISTER, LOOP ON THE CURRENTLY EXECUTING TEST, LOOP ON A SPECIFIC TEST, PERFORM ITERATIONS OF EACH TEST, AND SET UP ADDRESSES FOR POSSIBLE LOOPING ON ERRORS. IMPORTANT VALUES USED IN THIS ROUTINE ARE:

SMXCNT - MAXIMUM NUMBER OF ITERATIONS PER TEST

587  
588  
589  
590  
591  
592  
593  
594  
595  
596  
597  
598  
599  
600  
601  
602  
603  
604  
605  
606  
607  
608  
609  
610  
611  
612  
613  
614  
615  
616  
617  
618  
619  
620  
621  
622  
623  
624  
625  
626  
627  
628  
629  
630  
631  
632  
633  
634  
635  
636  
637  
638  
639  
640  
641  
642

643  
644  
645  
646  
647  
648  
649  
650  
651  
652  
653  
654  
655  
656  
657  
658  
659  
660  
661  
662  
663  
664  
665  
666  
667  
668  
669  
670  
671  
672  
673  
674  
675  
676  
677  
678  
679  
680  
681  
682  
683  
684  
685  
686  
687  
688  
689  
690  
691  
692  
693  
694  
695  
696  
697  
698

(GENERALLY WILL BE 2000(10))  
 STSTNM - A COUNTER INDICATING THE NUMBER (1-377(8)) OF  
 THE TEST CURRENTLY BEING EXECUTED  
 SLPADR - CONTAINS THE ADDRESS TO WHICH THE SCOPE  
 ROUTINE 10240 WILL LOOP, IF THE CURRENT TEST  
 IS BEING LOOPED UPON  
 SLPERR - CONTAINS THE ADDRESS TO WHICH THE ERROR  
 ROUTINE (SEE 9.3.3) WILL LOOP, IF AN ERROR  
 OCCURS AND THE LOOPING ON AN ERROR OPTION IS  
 SPECIFIED IN THE SWITCHES. SET UP BY SCOPE,  
 GENERALLY WILL BE THE SAME AS SLPADR, ABOVE.

## 9.3.3 ERROR ROUTINE - SERROR

THE ERROR ROUTINE IS ENTERED WHEN THE TEST CODE HAS DETERMINED THAT AN ERROR HAS OCCURRED AS PART OF A TEST. THROUGH USE OF THIS ROUTINE, THE TEST HAS A MEANS OF SIGNALING AN ERROR TO THE 10420 OPERATOR/MONITOR; AND IMPLEMENTING THE CONTROL FUNCTIONS FOR HALTING ON ERROR, BELL ON ERROR, AND LOOPING ON ERROR. IN ADDITION, THE ERROR ROUTINE HAS THE PROVISION TO TYPE OUT ON THE OPERATOR'S CONSOLE A MESSAGE BRIEFLY EXPLAINING THE ERROR, AND SOME OF THE MOST PERTINENT DATA VALUES TO HELP DIAGNOSE THE CAUSE (SEE SECTION 6.2).

THE CALLING MECHANISM IS SIMILAR TO THAT EMPLOYED FOR THE SCOPE ROUTINE (VIA A TRAP). EXCEPT IN THIS INSTANCE, THE "EMT" INSTRUCTION IS USED, TRAPPING THROUGH LOCATION 30(8). (NOTE THE EQUATE ERROR N=EMT N). THE LOWER BYTE OF THE EMT INSTRUCTION IS CAPABLE OF TRANSMITTING A NUMBER FROM 0-377(8), WHICH WILL BE TERMED THE "ERROR ITEM NUMBER." THIS NUMBER DETERMINES WHICH ERROR MESSAGE, AND ASSOCIATED DATA VALUES WILL BE TYPED OUT WHEN A PARTICULAR ERROR IS SIGNALLED. IF THIS NUMBER IS ZERO, JUST THE PC OF THE CALLING "ERROR" INSTRUCTION WILL BE TYPED, OTHERWISE, THE NUMBER IS USED AS AN INDEX THROUGH THE ERROR TABLE (SERRTB) TO FIND THE APPROPRIATE VALUES TO TYPE (SEE PROGRAM LISTING FOR FURTHER DETAILS).

IMPORTANT VALUES USED IN THIS ROUTINE ARE:

EREG0 THRU EREG7 - CONTENTS OF GENERAL REGISTERS R0  
 THRU R7 JUST BEFORE ERROR CALL  
 SERTTL - CUMULATIVE NUMBER OF ERRORS ENCOUNTERED TO  
 DATE  
 SERRPC - CONTAINS THE PC OF THE "ERROR" INSTRUCTION  
 JUST EXECUTED  
 SLPERR - CONTAINS THE ADDRESS WHICH WILL BE LOOPED  
 UPON FOR THE ERROR LOOPING FACILITY

## 9.3.4 ERROR MESSAGE TIMEOUT ROUTINE - STYPERR

THIS ROUTINE (STYPERR ENTRY POINT) IS CALLED BY THE ERROR PROCESSING ROUTINE DESCRIBED IN 9.3.3 ABOVE. ITS PURPOSE IS TO IMPLEMENT THE ERROR MESSAGE/DATA VALUE ERROR TIMEOUT

699  
700  
701  
702  
703  
704  
705  
706  
707  
708  
709  
710  
711  
712  
713  
714  
715  
716  
717  
718  
719  
720  
721  
722  
723  
724  
725  
726  
727  
728  
729  
730  
731  
732  
733  
734  
735  
736  
737  
738  
739  
740  
741  
742  
743  
744  
745  
746  
747  
748  
749  
750  
751  
752  
753  
754

FACILITY. THE SUBROUTINE WILL, GIVEN THE INDEXING BYTE FROM THE ERROR CALL INSTRUCTION, PICK UP THE CORRECT ERROR MESSAGE VECTOR FROM SEVRTB (ERROR TABLE), AND TYPE OUT THE ERROR MESSAGE, DATA HEADER, AND DATA VALUES ON THE CONSOLE.

## 9.3.5 TYPE ROUTINE - STYPE

THIS ROUTINE IS THE STANDARD SYSTEM TYPEOUT ROUTINE FOR ASCII SINGLE-CHARACTER-PER-BYTE STRINGS. IT IS CALLED THROUGH A TRAP INSTRUCTION WITH THE NEXT WORD CONTAINING THE ADDRESS OF THE FIRST CHARACTER IN THE STRING. TYPING TERMINATES WHEN AN ALL-ZERO BYTE IS FOUND. HORIZONTAL TAB STOPS ARE ALSO AUTOMATICALLY PLACED.

## 9.3.6 OCTAL NUMBER TYPE ROUTINE - STYPOC

THIS ROUTINE CONVERTS THE TOP NUMBER ON THE STACK TO A 6-DIGIT OCTAL REPRESENTATION, AND TYPES IT ON THE CONSOLE USING THE TYPE ROUTINE STYPE. SEE LISTING FOR OPTIONS AND FURTHER DETAILS.

## 9.3.7 POWER UP AND DOWN ROUTINES - SPWRUP AND SPWRDN

THESE TWO ROUTINES ARE ENTERED FOR THE POWER UP AND DOWN CONDITIONS, RESPECTIVELY. THE POWER DOWN ROUTINE (SPWRDN) SAVES THE GENERAL REGISTERS AND STACK POINTER. THE POWER UP ROUTINE (SPWRUP) CORRESPONDINGLY RESTORES THE REGISTERS, STACK POINTER, AND TYPES THE MESSAGE "POWER" WHEN POWER IS RESTORED. THE VOLATILE INTERNAL SWITCH REGISTER IS ALSO SAVED/RESTORED BY THIS ROUTINE.

## 9.3.8 END OF PASS ROUTINE - SEOP

THE END OF PASS ROUTINE COUNTS THE NUMBER OF PASSES PERFORMED, DINGS THE BELL/TYPES A MESSAGE (IF ENABLED), SETS/CLEARs THE T-BIT (IF ENABLED), AND ALSO INTERFACES TO THE MONITOR, IF PRESENT. IT ALSO OPTIONALLY LOOPS FOR A NUMBER OF SUBPASSES BEFORE SIGNALLING AN END OF PASS CONDITION.

## 10. ACT/APT/XXDP

## 10.1 ACT COMPATIBILITY

THIS PROGRAM WILL RUN UNDER THE ACT SYSTEM.

## 10.2 APT COMPATIBILITY

THIS PROGRAM WILL RUN UNDER THE APT SYSTEM MONITOR. ALL NECESSARY SOFTWARE COMMUNICATION HOOKS ARE PRESENT.

## 10.3 XXDP COMPATIBILITY

C02

FPU BASIC INSTR TESTS MACY11 27(1006) 09-FEB-77 09:48 PAGE 15  
DQFPA.NEM 09-FEB-77 09:46

755  
756  
757  
758  
759  
760  
761  
762  
763

MAINDEC-11-DQFPA-A

PAGE 15

FOR XXDP MEDIA COMPATIBILITY, THE TOP 2K WORDS OF THE 16K WORD  
MINIMUM MEMORY AREA ARE NOT DISTURBED DURING EXECUTION.

.ENDR



```

764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819

```

```

.TITLE FPU BASIC INSTR TESTS
.*COPYRIGHT (C) 1976
.*DIGITAL EQUIPMENT CORP.
.*MAYNARD, MASS. 01754
.*
.*PROGRAM BY DONALD NORTH
.*
.*THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC
.*PACKAGE (MAINDEC-11-DZQAC-C3), JAN 19, 1977.
.*

.SBTTL OPERATIONAL SWITCH SETTINGS
.*
.*      SWITCH  OCTAL          USE
.*      -----  -----  -----
.*      15      100000        HALT ON ERROR
.*      14      040000        LOOP ON CURRENTLY EXECUTING TEST
.*      13      020000        INHIBIT ERROR TYPEOUTS
.*      12      010000        INHIBIT STATUS TYPEOUTS
.*      11      004000        INHIBIT ITERATIONS
.*      10      000000        0=BELL ON PASS END
.*                          1=BELL ON ERROR
.*      9       001000        LOOP ON ERROR
.*      8       000400        LOOP ON TEST NUMBER IN "SLPTST"
.*      1       000000        0=TEST HFP/WFP ALTERNATELY EACH PASS
.*                          1=TEST ONLY UNIT SPECIFIED IN SW<00>
.*      0       000002        0=SELECT HFP, IF SW<01>=1
.*                          1=SELECT WFP, IF SW<01>=1
.*

.SBTTL BASIC DEFINITIONS

.*INITIAL ADDRESS OF THE STACK POINTER *** 1100 ***
STACK= 1100
.EQUIV EMT,ERROR      ;;BASIC DEFINITION OF ERROR CALL
.EQUIV IOT,SCOPE      ;;BASIC DEFINITION OF SCOPE CALL

.*MISCELLANEOUS DEFINITIONS
HT= 11      ;;CODE FOR HORIZONTAL TAB
LF= 12      ;;CODE FOR LINE FEED
CR= 15      ;;CODE FOR CARRIAGE RETURN
CRLF= 200   ;;CODE FOR CARRIAGE RETURN-LINE FEED
PS= 177776  ;;PROCESSOR STATUS WORD
.EQUIV PS,PSW
STKLMT= 177774 ;;STACK LIMIT REGISTER
PIRQ= 177772  ;;PROGRAM INTERRUPT REQUEST REGISTER
DSWR= 177570  ;;HARDWARE SWITCH REGISTER
DDISP= 177570 ;;HARDWARE DISPLAY REGISTER

.*GENERAL PURPOSE REGISTER DEFINITIONS
R0= %0      ;;GENERAL REGISTER
R1= %1      ;;GENERAL REGISTER
R2= %2      ;;GENERAL REGISTER
R3= %3      ;;GENERAL REGISTER
R4= %4      ;;GENERAL REGISTER
R5= %5      ;;GENERAL REGISTER

```

820	000006	R6=	%6	:: GENERAL REGISTER
821	000007	R7=	%7	:: GENERAL REGISTER
822	000006	SP=	%6	:: STACK POINTER
823	000007	PC=	%7	:: PROGRAM COUNTER
824				
825		:*PRIORITY LEVEL DEFINITIONS		
826	000000	PR0=	0	:: PRIORITY LEVEL 0
827	000040	PR1=	40	:: PRIORITY LEVEL 1
828	000100	PR2=	100	:: PRIORITY LEVEL 2
829	000140	PR3=	140	:: PRIORITY LEVEL 3
830	000200	PR4=	200	:: PRIORITY LEVEL 4
831	000240	PR5=	240	:: PRIORITY LEVEL 5
832	000300	PR6=	300	:: PRIORITY LEVEL 6
833	000340	PR7=	340	:: PRIORITY LEVEL 7
834				
835		:*"SWITCH REGISTER" SWITCH DEFINITIONS		
836	100000	SW15=	100000	
837	040000	SW14=	40000	
838	020000	SW13=	20000	
839	010000	SW12=	10000	
840	004000	SW11=	4000	
841	002000	SW10=	2000	
842	001000	SW09=	1000	
843	000400	SW08=	400	
844	000200	SW07=	200	
845	000100	SW06=	100	
846	000040	SW05=	40	
847	000020	SW04=	20	
848	000010	SW03=	10	
849	000004	SW02=	4	
850	000002	SW01=	2	
851	000001	SW00=	1	
852		.EQUIV	SW09, SW9	
853		.EQUIV	SW08, SW8	
854		.EQUIV	SW07, SW7	
855		.EQUIV	SW06, SW6	
856		.EQUIV	SW05, SW5	
857		.EQUIV	SW04, SW4	
858		.EQUIV	SW03, SW3	
859		.EQUIV	SW02, SW2	
860		.EQUIV	SW01, SW1	
861		.EQUIV	SW00, SW0	
862				
863		:*DATA BIT DEFINITIONS (BIT00 TO BIT15)		
864	100000	BIT15=	100000	
865	040000	BIT14=	40000	
866	020000	BIT13=	20000	
867	010000	BIT12=	10000	
868	004000	BIT11=	4000	
869	002000	BIT10=	2000	
870	001000	BIT09=	1000	
871	000400	BIT08=	400	
872	000200	BIT07=	200	
873	000100	BIT06=	100	
874	000040	BIT05=	40	
875	000020	BIT04=	20	

```

876      000010      BIT03= 10
877      000004      BIT02= 4
878      000002      BIT01= 2
879      000001      BIT00= 1
880      .EQUIV BIT09,BIT9
881      .EQUIV BIT08,BIT8
882      .EQUIV BIT07,BIT7
883      .EQUIV BIT06,BIT6
884      .EQUIV BIT05,BIT5
885      .EQUIV BIT04,BIT4
886      .EQUIV BIT03,BIT3
887      .EQUIV BIT02,BIT2
888      .EQUIV BIT01,BIT1
889      .EQUIV BIT00,BIT0
890
891      ;#BASIC "CPU" TRAP VECTOR ADDRESSES
892      000004      ERRVEC= 4      ; TIME OUT AND OTHER ERRORS
893      000010      RESVEC= 10     ; RESERVED AND ILLEGAL INSTRUCTIONS
894      000014      TBITVEC=14     ; "T" BIT
895      000014      TRTVEC= 14     ; TRACE TRAP
896      000014      BPTVEC= 14     ; BREAKPOINT TRAP (BPT)
897      000020      IOTVEC= 20     ; INPUT/OUTPUT TRAP (IOT) **SCOPE**
898      000024      PWRVEC= 24     ; POWER FAIL
899      000030      EMTVEC= 30     ; EMULATOR TRAP (EMT) **ERROR**
900      000034      TRAPVEC=34     ; "TRAP" TRAP
901      000060      TKVEC= 60      ; TTY KEYBOARD VECTOR
902      000064      TPVEC= 64      ; TTY PRINTER VECTOR
903      000240      PIRQVEC=240    ; PROGRAM INTERRUPT REQUEST VECTOR
904
905      ;#MED INSTR CODES
906      076600      MED= 076600    ; OPCODE
907
908      000022      RWHAMI= 022     ; READ WHAMI
909
910      000144      RFLAG= 144      ; READ FLAGS
911      000344      WFLAG= 344     ; WRITE FLAGS
912
913      ;#FLOATING POINT INTERRUPT VECTOR
914      000244      FPPVEC= 244
915
916      ;#FLOATING POINT REGISTER DEFINITIONS
917      000000      ACO= %0
918      000001      AC1= %1
919      000002      AC2= %2
920      000003      AC3= %3
921      000004      AC4= %4
922      000005      AC5= %5
923
924      ;#BIT PATTERNS FOR TESTS
925      052525      ALTP= 052525    ; 0101...01
926      125252      ALTN= 125252    ; 1010...10
927      007417      ALT4P= 007417   ; 0000111100001111
928      170360      ALT4N= 170360   ; 1111000011110000
929      177777      M1= 177777      ; 1111...11 MINUS ONE, ALL 1'S
930      100000      M0= 100000      ; 1000...00 MINUS ZERO
931      077777      LGP= 077777    ; 0111...11 LGST + NUM (1ST WD FLT )
  
```

932 177777  
 933 000200  
 934 100200  
 935 000177  
 936 100177  
 937 040200  
 938 140200  
 939  
 940  
 941 147757  
 942 000000  
 943  
 944  
 945 177760  
 946  
 947  
 948 104117  
 949  
 950  
 951  
 952 000000  
 953  
 954  
 955  
 956 000174  
 957 000174 000000  
 958 000176 000000  
 959  
 960 000200 000137 002506  
 961  
 962  
 963  
 964  
 965  
 966 000204  
 967 000046 000046  
 968 000046 020510  
 969 000052 000052  
 970 000052 000000  
 971 000204  
 972 001000  
 973  
 974  
 975  
 976  
 977  
 978 001000  
 979 000024 000024  
 980 000024 000200  
 981 000044 000044  
 982 000044 001000  
 983 001000  
 984  
 985  
 986  
 987

```

LGN= 177777 ; 1111...11 LGST - NUM (1ST WD FLT)
SMP= 000200 ; +1*2**-128, SMLT + NUM (1ST WD FLT)
SMN= 100200 ; -1*2**-128, SMLT - NUM (1ST WD FLT)
ZXIMP= 000177 ; ZERO EXP, ALL 1-S MANT (1ST WD FLT)
ZXIMN= 100177 ; ZERO EXP, ALL 1-S MANT (1ST WD FLT)
FIP= 040200 ; +1.0E+0, 1ST WD FLT
FIN= 140200 ; -1.0E+0, 1ST WD FLT

;#FPS BIT PATTERNS
FPS1= 147757 ; ALL BITS ON (READABLE)
FPS0= 000000 ; ALL BITS OFF

;#PSW BIT PATTERNS
CCONLY= 177760 ; FOR BIC TO GET CC BITS ONLY

;MISC
ERROR117= ERROR!117 ; FOR USE AS A LITERAL

.SBTTL TRAP CATCHER
.=0
;#ALL UNUSED LOCATIONS FROM 4 - 776 CONTAIN A ".+2,HALT"
;#SEQUENCE TO CATCH ILLEGAL TRAPS AND INTERRUPTS
;#LOCATION 0 CONTAINS 0 TO CATCH IMPROPERLY LOADED VECTORS
.=174
DISPREG: .WORD 0 ;;SOFTWARE DISPLAY REGISTER
SWREG: .WORD 0 ;;SOFTWARE SWITCH REGISTER
.SBTTL STARTING ADDRESS(ES)
JMP @START ;;JUMP TO STARTING ADDRESS OF PROGRAM

.SBTTL ACT11 HOOKS
;*****
;HOOKS REQUIRED BY ACT11
$SVPC=. ;SAVE PC
.=46
$ENDAD ;;1)SET LOC.46 TO ADDRESS OF $ENDAD IN .$EOP
.=52
.WORD 0 ;;2)SET LOC.52 TO ZERO
.$SVPC ;; RESTORE PC
.=1000

.SBTTL APT PARAMETER BLOCK
;*****
;SET LOCATIONS 24 AND 44 AS REQUIRED FOR APT
;*****
.$X=. ;;SAVE CURRENT LOCATION
.=24 ;;SET POWER FAIL TO POINT TO START OF PROGRAM
200 ;;FOR APT START UP
.=44 ;;POINT TO APT INDIRECT ADDRESS PNTR.
$APTHDR ;;POINT TO APT HEADER BLOCK
.=.$X ;;RESET LOCATION COUNTER
;*****
;SETUP APT PARAMETER BLOCK AS DEFINED IN THE APT-PDP11 DIAGNOSTIC
;INTERFACE SPEC.
  
```

H02

FPU BASIC INSTR TESTS MACY11 27(1006) 09-FEB-77 09:48 PAGE 22  
DQFPAA.P11 09-FEB-77 09:42 APT PARAMETER BLOCK

988	001000		\$APTHD:		
989	001000	000000	\$HIBTS:	.WORD	0 ;: TWO HIGH BITS OF 18 BIT MAILBOX ADDR.
990	001002	001324	\$MADR:	.WORD	\$MAIL ;: ADDRESS OF APT MAILBOX (BITS 0-15)
991	001004	000001	\$STMT:	.WORD	1 ;: RUN TIM OF LONGEST TEST
992	001006	000001	\$PASTM:	.WORD	1 ;: RUN TIME IN SECS. OF 1ST PASS ON 1 UNIT (QUICK VERIFY)
993	001010	000000	\$UNITM:	.WORD	0 ;: ADDITIONAL RUN TIME (SECS) OF A PASS FOR EACH ADDITIONAL UNIT
994	001012	000014		.WORD	SETEND-\$MAIL/2 ;: LENGTH MAILBOX-ETABLE(WORDS)
995					

996  
997  
998  
999  
1000  
1001  
1002 001100  
1003 001100 000000  
1004 001100 000000  
1005 001102 000000  
1006 001104 000000  
1007 001106 000000  
1008 001110 000000  
1009 001112 000000  
1010 001114 000000  
1011 001116 000000  
1012 001120 000000  
1013 001122 000001  
1014 001124 000000  
1015 001126 000000  
1016 001130 000000  
1017 001132 000000  
1018 001134 000000  
1019 001136 000000  
1020 001140 000000  
1021 001142 000  
1022 001143 000  
1023 001144 000000  
1024 001146 177570  
1025 001150 177570  
1026 001152 177560  
1027 001154 177562  
1028 001156 177564  
1029 001160 177566  
1030 001162 000  
1031 001163 002  
1032 001164 012  
1033 001165 000  
1034 001166 000000  
1035  
1036 001170 000000  
1037 001172 000000  
1038 001174 000000  
1039 001176 000000  
1040 001200 000000  
1041 001202 000000  
1042 001204 000000  
1043 001206 000000  
1044 001210 000000  
1045 001212 000000  
1046 001214 000000  
1047 001216 000000  
1048 001220 000000  
1049 001222 000000  
1050 001224 000000  
1051 001226 000000

.SBTTL COMMON TAGS

\*\*\*\*\*  
\*THIS TABLE CONTAINS VARIOUS COMMON STORAGE LOCATIONS  
\*USED IN THE PROGRAM.

SCMTAG: . =1100

.WORD 0  
STSTNM: .WORD 0  
SERFLG: .WORD 0  
SICNT: .WORD 0  
SLPADR: .WORD 0  
SLPTST: .WORD 0  
SLPERR: .WORD 0  
SERTTL: .WORD 0  
SITEMB: .WORD 0  
SERMAX: .WORD 1  
SERRPC: .WORD 0  
SGDADR: .WORD 0  
SBDADR: .WORD 0  
SGDDAT: .WORD 0  
SBDDAT: .WORD 0  
SAUTOB: .BYTE 0  
SINTAG: .BYTE 0  
SWR: .WORD DSWR  
DISPLAY: .WORD DDISP  
STKS: 177560  
STKB: 177562  
STPS: 177564  
STPB: 177566  
SNLL: .BYTE 0  
SFILLS: .BYTE 2  
SFILLC: .BYTE 12  
STPFLG: .BYTE 0  
SREGAD: .WORD 0  
SREG0: .WORD 0  
SREG1: .WORD 0  
SREG2: .WORD 0  
SREG3: .WORD 0  
SREG4: .WORD 0  
SREG5: .WORD 0  
SREG6: .WORD 0  
SREG7: .WORD 0  
SREG10: .WORD 0  
SREG11: .WORD 0  
SREG12: .WORD 0  
SREG13: .WORD 0  
SREG14: .WORD 0  
SREG15: .WORD 0  
SREG16: .WORD 0  
SREG17: .WORD 0

;; START OF COMMON TAGS  
;; CONTAINS THE TEST NUMBER  
;; CONTAINS ERROR FLAG  
;; CONTAINS SUBTEST ITERATION COUNT  
;; CONTAINS SCOPE LOOP ADDRESS  
;; CONTAINS TEST NUMBER TO LOOP UPON  
;; CONTAINS SCOPE RETURN FOR ERRORS  
;; CONTAINS TOTAL ERRORS DETECTED  
;; CONTAINS ITEM CONTROL BYTE  
;; CONTAINS MAX. ERRORS PER TEST  
;; CONTAINS PC OF LAST ERROR INSTRUCTION  
;; CONTAINS ADDRESS OF 'GOOD' DATA  
;; CONTAINS ADDRESS OF 'BAD' DATA  
;; CONTAINS 'GOOD' DATA  
;; CONTAINS 'BAD' DATA  
;; RESERVED--NOT TO BE USED  
;; AUTOMATIC MODE INDICATOR  
;; INTERRUPT MODE INDICATOR  
;; ADDRESS OF SWITCH REGISTER  
;; ADDRESS OF DISPLAY REGISTER  
;; TTY KBD STATUS  
;; TTY KBD BUFFER  
;; TTY PRINTER STATUS REG. ADDRESS  
;; TTY PRINTER BUFFER REG. ADDRESS  
;; CONTAINS NULL CHARACTER FOR FILLS  
;; CONTAINS # OF FILLER CHARACTERS REQUIRED  
;; INSERT FILL CHARS. AFTER A "LINE FEED"  
;; "TERMINAL AVAILABLE" FLAG (BIT<07>=0=YES)  
;; CONTAINS THE ADDRESS FROM  
;; WHICH (SREG0) WAS OBTAINED  
;; CONTAINS ((SREGAD)+0)  
;; CONTAINS ((SREGAD)+2)  
;; CONTAINS ((SREGAD)+4)  
;; CONTAINS ((SREGAD)+6)  
;; CONTAINS ((SREGAD)+10)  
;; CONTAINS ((SREGAD)+12)  
;; CONTAINS ((SREGAD)+14)  
;; CONTAINS ((SREGAD)+16)  
;; CONTAINS ((SREGAD)+20)  
;; CONTAINS ((SREGAD)+22)  
;; CONTAINS ((SREGAD)+24)  
;; CONTAINS ((SREGAD)+26)  
;; CONTAINS ((SREGAD)+30)  
;; CONTAINS ((SREGAD)+32)  
;; CONTAINS ((SREGAD)+34)  
;; CONTAINS ((SREGAD)+36)

1052 001230 000000  
 1053 001232 000000  
 1054 001234 000000  
 1055 001236 000000  
 1056 001240 000000  
 1057 001242 000000  
 1058 001244 000000  
 1059 001246 000000  
 1060 001250 000000  
 1061 001252 000000  
 1062 001254 000000  
 1063 001256 000000  
 1064 001260 000000  
 1065 001262 000000  
 1066 001264 000000  
 1067 001266 000000  
 1068 001270 000000  
 1069 001272 000000  
 1070 001274 000000  
 1071 001276 000000  
 1072 001300 000000  
 1073 001302 000000  
 1074 001304 000000  
 1075 001306 000000  
 1076 001310 000000  
 1077 001312 000000  
 1078 001314 177607 000377  
 1079 001320 077  
 1080 001321 015  
 1081 001322 000012  
 1082  
 1083  
 1084  
 1085  
 1086  
 1087 001324  
 1088 001324 000000  
 1089 001326 000000  
 1090 001330 000000  
 1091 001332 000000  
 1092 001334 000000  
 1093 001336 000000  
 1094 001340 000000  
 1095 001342 000000  
 1096 001344  
 1097 001344 000  
 1098 001345 000  
 1099 001346 000000  
 1100 001350 000000  
 1101 001352 000000  
 1102  
 1103  
 1104  
 1105  
 1106  
 1107

STMP0: .WORD 0  
 STMP1: .WORD 0  
 STMP2: .WORD 0  
 STMP3: .WORD 0  
 STMP4: .WORD 0  
 STMP5: .WORD 0  
 STMP6: .WORD 0  
 STMP7: .WORD 0  
 STMP10: .WORD 0  
 STMP11: .WORD 0  
 STMP12: .WORD 0  
 STMP13: .WORD 0  
 STMP14: .WORD 0  
 STMP15: .WORD 0  
 STMP16: .WORD 0  
 STMP17: .WORD 0  
 STMP20: .WORD 0  
 STMP21: .WORD 0  
 STMP22: .WORD 0  
 STMP23: .WORD 0  
 STMP24: .WORD 0  
 STMP25: .WORD 0  
 STMP26: .WORD 0  
 STMP27: .WORD 0  
 STIMES: 0  
 \$ESCAPE: 0  
 \$BELL: .ASCIZ <207><377><377>  
 \$QUES: .ASCII /?/  
 \$CRLF: .ASCII <15>  
 \$LF: .ASCIZ <12>

:: USER DEFINED  
 :: USER DEFINED  
 :: USER DEFINED  
 :: USER DEFINED  
 :: USER DEFINED  
 :: USER DEFINED  
 :: USER DEFINED  
 :: USER DEFINED  
 :: USER DEFINED  
 :: USER DEFINED  
 :: USER DEFINED  
 :: USER DEFINED  
 :: USER DEFINED  
 :: USER DEFINED  
 :: USER DEFINED  
 :: USER DEFINED  
 :: USER DEFINED  
 :: USER DEFINED  
 :: USER DEFINED  
 :: USER DEFINED  
 :: USER DEFINED  
 :: USER DEFINED  
 :: USER DEFINED  
 :: USER DEFINED  
 :: USER DEFINED  
 :: USER DEFINED  
 :: USER DEFINED  
 :: USER DEFINED  
 :: USER DEFINED  
 :: MAX. NUMBER OF ITERATIONS  
 :: ESCAPE ON ERROR ADDRESS  
 :: CODE FOR BELL  
 :: QUESTION MARK  
 :: CARRIAGE RETURN  
 :: LINE FEED

:: \*\*\*\*\*  
 .SBTTL APT MAILBOX-ETABLE

:: \*\*\*\*\*

.EVEN  
 \$MAIL: :: APT MAILBOX  
 \$MSGTY: .WORD AMSGTY :: MESSAGE TYPE CODE  
 \$FATAL: .WORD AFATAL :: FATAL ERROR NUMBER  
 \$TESTN: .WORD ATESTN :: TEST NUMBER  
 \$PASS: .WORD APASS :: PASS COUNT  
 \$DEVCT: .WORD ADEVCT :: DEVICE COUNT  
 \$UNIT: .WORD AUNIT :: I/O UNIT NUMBER  
 \$MSGAD: .WORD AMSGAD :: MESSAGE ADDRESS  
 \$MSGLG: .WORD AMSGLG :: MESSAGE LENGTH  
 \$ETABLE: :: APT ENVIRONMENT TABLE  
 \$ENV: .BYTE AENV :: ENVIRONMENT BYTE  
 \$ENVM: .BYTE AENVM :: ENVIRONMENT MODE BITS  
 \$SWREG: .WORD ASWREG :: APT SWITCH REGISTER  
 \$USWR: .WORD AUSWR :: USER SWITCHES  
 \$CPUOP: .WORD ACPUOP :: CPU TYPE, OPTIONS

BIT 15-11=CPU TYPE  
 11/04=01, 11/05=02, 11/20=03, 11/40=04, 11/45=05  
 11/70=06, PDQ=07, Q=10  
 BIT 10=REAL TIME CLOCK  
 BIT 9=FLOATING POINT PROCESSOR  
 BIT 8=MEMORY MANAGEMENT

K02

FPU BASIC INSTR TESTS MACY11 27(1006) 09-FEB-77 09:48 PAGE 25  
DQFPA.P11 09-FEB-77 09:42 APT MAILBOX-ETABLE

1108 001354  
1109

SETEND:  
.MEXIT



1110 .SBTTL ERROR POINTER TABLE

1111  
1112 001354 SERRTB:

1113  
1114 : \*THIS TABLE CONTAINS THE INFORMATION FOR EACH ERROR THAT CAN OCCUR.  
1115 : \*THE INFORMATION IS OBTAINED BY USING THE INDEX NUMBER FOUND IN  
1116 : \*LOCATION SITEMB. THIS NUMBER INDICATES WHICH ITEM IN THE TABLE IS PERTINENT.  
1117 : \*NOTE1: IF SITEMB IS 0 THE ONLY PERTINENT DATA IS (SERRPC).  
1118 : \*NOTE2: EACH ITEM IN THE TABLE CONTAINS 4 POINTERS EXPLAINED AS FOLLOWS:  
1119

1120 : \* EM :POINTS TO THE ERROR MESSAGE  
1121 : \* DH :POINTS TO THE DATA HEADER  
1122 : \* DT :POINTS TO THE DATA  
1123 : \* DF :POINTS TO THE DATA FORMAT

1124 : \*NOTE: ERROR VECTOR TABLE (SERRTB) HAS BEEN MODIFIED,  
1125 : \* ELIMINATING UNUSED VALUE FOR DATA FORMAT POINTER.  
1126 : \* ERROR TYPING ROUTINE HAS ALSO BEEN MODIFIED  
1127 : \* ACCORDINGLY.

1128 : \* \* \* \* \* SPECIFIC TEST VECTORS \* \* \* \* \*

1129 001354 026023 030215 032620 EMV001: .WORD EMF,DHA,DTB : FPS RIPPLE 1  
1130 001362 026060 030215 032620 EMV002: .WORD EMG,DHA,DTB : FPS RIPPLE 0  
1131 001370 026115 030233 032614 EMV003: .WORD EMH,DHB,DTA : SETF  
1132 001376 026154 030233 032614 EMV004: .WORD EMI,DHB,DTA : SETD  
1133 001404 026211 030233 032614 EMV005: .WORD EMJ,DHB,DTA : SETI  
1134 001412 026250 030233 032614 EMV006: .WORD EMK,DHB,DTA : SETL  
1135 : \* \* \* \* \* SRC/DST ADDR MODES VECTORS \* \* \* \* \*  
1136 001420 026305 030765 033104 EMV007: .WORD EML,DHO,DTT : M05-M77 RESULT  
1137 001426 026477 031054 033132 EMV010: .WORD EMN,DHO,DTV : M13-M35 DSTREG  
1138 001434 026305 031054 033132 EMV011: .WORD EML,DHO,DTV : M13-M35 RESULT  
1139 001442 026375 031016 033116 EMV012: .WORD EMN,DHP,DTU : M21-M53 SRCREG  
1140 001450 026477 031016 033116 EMV013: .WORD EMN,DHP,DTU : M21-M53 DSTREG  
1141 001456 026305 031016 033116 EMV014: .WORD EML,DHP,DTU : M21-M53 RESULT, M63-M71 RESULT  
1142 001464 026375 030652 033046 EMV015: .WORD EMN,DHL,DTQ : M30-M57 SRCREG  
1143 001472 026305 030652 033046 EMV016: .WORD EML,DHL,DTQ : M30-M57 RESULT  
1144 001500 026375 030622 033034 EMV017: .WORD EMN,DHK,DTP : M42-M00 SRCREG  
1145 001506 026305 030622 033034 EMV020: .WORD EML,DHK,DTP : M42-M00 RESULT  
1146 001514 026375 030734 033072 EMV021: .WORD EMN,DHN,DTS : M54-M37 SRCREG  
1147 001522 026305 030734 033072 EMV022: .WORD EML,DHN,DTS : M54-M37 RESULT, M77-M24 RESULT  
1148 001530 026305 030703 033060 EMV023: .WORD EML,DHN,DTR : M72-M27 RESULT  
1149 001536 026477 030577 033024 EMV024: .WORD EMN,DHJ,DT0 : M27-M43 DSTREG  
1150 001544 026305 030577 033024 EMV025: .WORD EML,DHJ,DT0 : M27-M43 RESULT  
1151 001552 026305 030554 033014 EMV026: .WORD EML,DHI,DTN : M37-M62 RESULT, M67-M12 RESULT  
1152 001560 026477 030734 033072 EMV027: .WORD EMN,DHN,DTS : M77-M24 DSTREG

1153 : \* \* \* \* \* VECTOR FOR CFCC TEST \* \* \* \* \*

1154 001566 026601 030215 032620 EMV030: .WORD EMO,DHA,DTB : CFCC INSTR  
1155 : \* \* \* \* \* FSRC/FDST D ADDR MODES VECTORS \* \* \* \* \*  
1156 001574 026657 031215 033174 EMV031: .WORD EMP,DHT,DTY : M15-M67 RESULT, M67-M25 RESULT  
1157 001602 026753 031151 033160 EMV032: .WORD EMQ,DHS,DTX : M44-M37 SRCREG  
1158 001610 026657 031151 033160 EMV033: .WORD EMP,DHS,DTX : M44-M37 RESULT  
1159 001616 027062 031666 033336 EMV034: .WORD EMR,DHAA,DTAF : M75-M34 DSTREG  
1160 001624 026657 031666 033336 EMV035: .WORD EMP,DHAA,DTAF : M75-M34 RESULT  
1161 001632 027062 031215 033174 EMV036: .WORD EMR,DHT,DTY : M67-M25 DSTREG  
1162 001640 026753 031261 033210 EMV037: .WORD EMQ,DHU,DTZ : M20-M13 SRCREG  
1163 001646 026657 031261 033210 EMV040: .WORD EMP,DHU,DTZ : M20-M13 RESULT  
1164 001654 026753 031604 033316 EMV041: .WORD EMQ,DHZ,DTAE : M51-M77 SRCREG  
1165 001662 026657 031604 033316 EMV042: .WORD EMP,DHZ,DTAE : M51-M77 RESULT

1166	001670	026657	031403	033244	EMV043:	.WORD	EMP, DHW, DTAB	:	M27-M70	RESULT
1167	001676	026657	031531	033300	EMV044:	.WORD	EMP, DHY, DTAD	:	M77-M64	RESULT
1168	001704	026753	031456	033262	EMV045:	.WORD	EMQ, DHX, DTAC	:	M32-M27	SRCREG
1169	001712	026657	031456	033262	EMV046:	.WORD	EMP, DHX, DTAC	:	M32-M27	RESULT, M00-M52 RESULT
1170	001720	027062	031332	033226	EMV047:	.WORD	EMR, DHV, DTA	:	M63-M41	DSTREG
1171	001726	026657	031332	033226	EMV050:	.WORD	EMP, DHV, DTA	:	M63-M41	RESULT
1172	001734	026657	031112	033146	EMV051:	.WORD	EMP, DHR, DTW	:	M37-M03	RESULT
1173	001742	027062	031456	033262	EMV052:	.WORD	EMR, DHX, DTAC	:	M00-M52	DSTREG
1174						;	***** FSRC/FDST F ADDR	:	MODES VECTORS	*****
1175	001750	027374	032063	033410	EMV053:	.WORD	EMU, DHAD, DTAI	:	M12-M45	DSTREG
1176	001756	027171	032063	033410	EMV054:	.WORD	EMS, DHAD, DTAI	:	M12-M45	RESULT
1177	001764	027265	032177	033442	EMV055:	.WORD	EMT, DHAF, DTAK	:	M34-M60	SRCREG
1178	001772	027171	032177	033442	EMV056:	.WORD	EMS, DHAF, DTAK	:	M34-M60	RESULT
1179	002000	027265	032470	033542	EMV057:	.WORD	EMT, DHAJ, DTAO	:	M50-M32	SRCREG
1180	002006	027374	032470	033542	EMV060:	.WORD	EMU, DHAJ, DTAO	:	M50-M32	DSTREG
1181	002014	027171	032470	033542	EMV061:	.WORD	EMS, DHAJ, DTAO	:	M50-M32	RESULT
1182	002022	027171	032255	033462	EMV062:	.WORD	EMS, DHAG, DTAL	:	M72-M11	RESULT
1183	002030	027374	032020	033374	EMV063:	.WORD	EMU, DHAC, DTAH	:	M37-M23	DSTREG
1184	002036	027171	032020	033374	EMV064:	.WORD	EMS, DHAC, DTAH	:	M37-M23	RESULT, M23-M37 RESULT
1185	002044	027171	032411	033522	EMV065:	.WORD	EMS, DHAJ, DTAN	:	M77-M75	RESULT
1186	002052	027265	032020	033374	EMV066:	.WORD	EMT, DHAC, DTAH	:	M23-M37	SRCREG
1187	002060	027265	032333	033502	EMV067:	.WORD	EMT, DHAH, DTAM	:	M45-M54	SRCREG
1188	002066	027374	032333	033502	EMV070:	.WORD	EMU, DHAH, DTAM	:	M45-M54	DSTREG
1189	002074	027171	032333	033502	EMV071:	.WORD	EMS, DHAH, DTAM	:	M45-M54	RESULT
1190	002102	027171	032133	033426	EMV072:	.WORD	EMS, DHAE, DTAJ	:	M27-M77	RESULT
1191	002110	027171	032554	032626	EMV073:	.WORD	EMS, DHAK, DTC	:	M67-M67	RESULT, M02-M27 RESULT
1192	002116	027171	031755	033360	EMV074:	.WORD	EMS, DHAB, DTAG	:	M61-M01	RESULT
1193						;	***** FP ACC ALL THERE ?	:	VECTOR	*****
1194	002124	025752	030242	032626	EMV075:	.WORD	EME, DHC, DTC	:	FP AC ALL THERE ?	
1195						;	***** FPS ERROR VECTORS	:	*****	
1196	002132	025543	030300	032640	EMV076:	.WORD	ENA, DHD, DTD	:	FPS - ABS, NEG, CLR, TST F	
1197	002140	025543	030300	032646	EMV077:	.WORD	ENA, DHD, DTE	:	FPS - LDD/STD, ABS, NEG, CLR, TST D	
1198	002146	025543	030300	032654	EMV100:	.WORD	ENA, DHD, DTF	:	FPS - LDD/LDF/STD	
1199						;	***** FEC/FEA ERROR VECTORS	:	*****	
1200	002154	025567	030314	032662	EMV101:	.WORD	EMB, DHE, DTG	:	FEC/FEA - ABS, NEG, CLR, TST F	
1201	002162	025567	030314	032674	EMV102:	.WORD	EMB, DHE, DTH	:	FEC/FEA - LDD/STD, ABS, ..., CLR D	
1202	002170	025567	030314	032706	EMV103:	.WORD	EMB, DHE, DTI	:	FEC/FEA - LDD/LDF/STD	
1203						;	***** RESULT VECTORS	:	*****	
1204	002176	027503	030412	032732	EMV104:	.WORD	EMV, DHG, DTK	:	RESULT - LDD/STD	
1205	002204	027541	030412	032754	EMV105:	.WORD	EMW, DHG, DTL	:	RESULT - LDD/LDF/STD	
1206	002212	025617	030354	032720	EMV106:	.WORD	ENC, DHF, DTJ	:	RESULT - ABSF	
1207	002220	025617	030412	032732	EMV107:	.WORD	ENC, DHG, DTK	:	RESULT - ABSD	
1208	002226	027603	030354	032720	EMV110:	.WORD	EMX, DHF, DTJ	:	RESULT - NEGF	
1209	002234	027603	030412	032732	EMV111:	.WORD	EMX, DHG, DTK	:	RESULT - NEG0	
1210	002242	027737	030354	032720	EMV112:	.WORD	EMZ, DHF, DTJ	:	RESULT - CLRF	
1211	002250	027737	030412	032732	EMV113:	.WORD	EMZ, DHG, DTK	:	RESULT - CLRD	
1212	002256	027651	030354	032720	EMV114:	.WORD	EMY, DHF, DTJ	:	RESULT - TSTF	
1213	002264	027651	030412	032732	EMV115:	.WORD	EMY, DHG, DTK	:	RESULT - TSTD	
1214						;	***** ILLEGAL TRAP CATCHER VECTOR	:	*****	
1215	002272	025665	030510	032776	EMV116:	.WORD	END, DHH, DTH	:	ILLEGAL TRAP CATCHER	
1216						;	***** PC MODE 2 WRONG INCRE VECTORS	:	*****	
1217	002300	030005	032612	033564	EMV117:	.WORD	EMAA, DHAL, DTAP	:	+0	
1218	002306	030042	032612	033564	EMV120:	.WORD	EMAB, DHAL, DTAP	:	+4	
1219	002314	030113	032612	033564	EMV121:	.WORD	EMAC, DHAL, DTAP	:	+6	
1220	002322	030157	032612	033564	EMV122:	.WORD	EMAD, DHAL, DTAP	:	+10	

```

1221
1222
1223 002330 000000
1224 002332 000000
1225 002334 000000
1226 002336 000000
1227 002340 000000
1228 002342 000000
1229 002344 000000
1230
1231
1232 002346 000000
1233 002350 000000
1234 002352 000000
1235 002354 000000
1236 002356 000000
1237 002360 000000
1238 002362 000000
1239 002364 000000
1240
1241
1242 002366 052525 177777 125252
1243 002374 000000
1244
1245
1246
1247 002376 005015 005012 042115
1248 002404 030455 026461 050504
1249 002412 050106 026501 027101
1250 002420 027056
1251 002422 042120 026520 030461
1252 002430 033057 020130 027106
1253 002436 027120 027125 041040
1254 002444 051501 041511 044440
1255 002452 051516 051124 041525
1256 002460 044524 047117 052040
1257 002466 051505 051524 005015
1258 002474 000
1259 002475 015 050012 051501
1260 002502 020123 000043

.SBTTL PROGRAM DEFINED COMMON TAGS
: #VARIABLES
FPS: .WORD 0 ; FPS STORED HERE AFTER STFPS
FEC: .WORD 0 ; FEC STORED HERE AFTER STST
FEA: .WORD 0 ; FEA STORED HERE AFTER STST
FPPOPC: .WORD 0 ; OLD PC SAVED HERE AFTER TRAP
FPPOPS: .WORD 0 ; OLD PS SAVED HERE AFTER TRAP
FPPOSP: .WORD 0 ; SP AFTER TRAP
EXPFEA: .WORD 0 ; EXPECTED FEA

: #REGISTER CONTENTS, AT ERROR, STORED HERE
EREG0: .WORD 0
EREG1: .WORD 0
EREG2: .WORD 0
EREG3: .WORD 0
EREG4: .WORD 0
EREG5: .WORD 0
EREG6: .WORD 0
EREG7: .WORD 0

: #CONSTANTS
PREVAC: .WORD ALTP,M1,ALTN,0 ; PREV CONTENTS OF FLOAT AC

: #MESSAGES FOR BEGIN PROGRAM/START OF PASS/ETC
BGNMES: .ASCII <15><12><12><12>"MD-11-DGFPA-A..."

.ASCIZ "PDP-11/6X F.P.U. BASIC INSTRUCTION TESTS"<15><12>

NWPAS1: .ASCIZ <15><12>"PASS #"
```

```

1261 .SBTTL START OF PASS ROUTINE
1262
1263 .EVEN ; START ON AN EVEN BOUNDARY
1264
1265 ;;*****
1266 .ENABL AMA ; BEGIN ASSEMBLING RELATIVE REFERENCES AS ABSOLUTE
1267 ;;*****
1268
1269 002506 START:
1270 .SBTTL INITIALIZE THE COMMON TAGS
1271 ;;CLEAR THE COMMON TAGS ($CMTAG) AREA
1272 002506 012706 001100 MOV #SCMTAG,R6 ;;FIRST LOCATION TO BE CLEARED
1273 002512 005026 CLR (R6)+ ;;CLEAR MEMORY LOCATION
1274 002514 022706 001146 CMP #SWR,R6 ;;DONE?
1275 002520 001374 BNE .-6 ;;LOOP BACK IF NO
1276 002522 012706 001100 MOV #STACK,SP ;;SETUP THE STACK POINTER
1277 ;;INITIALIZE A FEW VECTORS
1278 002526 012737 023356 000020 MOV #SCOPE,#IOTVEC ;;IOT VECTOR FOR SCOPE ROUTINE
1279 002534 012737 000340 000022 MOV #340,#IOTVEC+2 ;;LEVEL 7
1280 002542 012737 023634 000030 MOV #ERROR,#EMTVEC ;;EMT VECTOR FOR ERROR ROUTINE
1281 002550 012737 000340 000032 MOV #340,#EMTVEC+2 ;;LEVEL 7
1282 002556 012737 025270 000034 MOV #STRAP,#TRAPVEC ;;TRAP VECTOR FOR TRAP CALLS
1283 002564 012737 000340 000036 MOV #340,#TRAPVEC+2 ;;LEVEL 7
1284 002572 012737 025336 000024 MOV #SPWRDN,#PMRVEC ;;POWER FAILURE VECTOR
1285 002600 012737 000340 000026 MOV #340,#PMRVEC+2 ;;LEVEL 7
1286 002606 013737 020460 020452 MOV SENDCT,SEOPCT ;;SETUP END-OF-PROGRAM COUNTER
1287 002614 005037 001310 CLR $TIMES ;;INITIALIZE NUMBER OF ITERATIONS
1288 002620 005037 001312 CLR $ESCAPE ;;CLEAR THE ESCAPE ON ERROR ADDRESS
1289 002624 012737 000001 001122 MOV #1,SERMAX ;;ALLOW ONE ERROR PER TEST
1290 002632 012737 002632 001110 MOV #.,$LPADR ;;INITIALIZE THE LOOP ADDRESS FOR SCOPE
1291 002640 012737 002640 001114 MOV #.,$LPERR ;;SETUP THE ERROR LOOP ADDRESS
1292 ;;SIZE FOR A HARDWARE SWITCH REGISTER. IF NOT FOUND OR IT IS
1293 ;;EQUAL TO A "-1" SETUP FOR A SOFTWARE SWITCH REGISTER.
1294 002646 013746 000004 MOV #ERRVEC,-(SP) ;;SAVE ERROR VECTOR
1295 002652 012737 002706 000004 MOV #64$,#ERRVEC ;;SET UP ERROR VECTOR
1296 002660 012737 177570 001146 MOV #DSWR,SWR ;;SETUP FOR A HARDWARE SWICH REGISTER
1297 002666 012737 177570 001150 MOV #DDISP,DISPLAY ;;AND A HARDWARE DISPLAY REGISTER
1298 002674 022777 177777 176244 CMP #-1,$SWR ;;TRY TO REFERENCE HARDWARE SWR
1299 002702 001012 BNE 66$ ;;BRANCH IF NO TIMEOUT TRAP OCCURRED
1300 ;;AND THE HARDWARE SWR IS NOT = -1
1301 002704 000403 BR 65$ ;;BRANCH IF NO TIMEOUT
1302 002706 012716 002714 64$: MOV #65$,(SP) ;;SET UP FOR TRAP RETURN
1303 002712 000002 RTI
1304 002714 012737 000176 001146 65$: MOV #SWREG,SWR ;;POINT TO SOFTWARE SWR
1305 002722 012737 000174 001150 MOV #DISPREG,DISPLAY
1306 002730 012637 000004 66$: MOV (SP)+,#ERRVEC ;;RESTORE ERROR VECTOR
1307
1308 002734 005037 001332 CLR $PASS ;;CLEAR PASS COUNT
1309 002740 132737 000200 001345 BITB #APTSIZE,$ENVM ;;TEST USER SIZE UNDER APT
1310 002746 001403 BEQ 67$ ;;YES,USE NON-APT SWITCH
1311 002750 012737 001346 001146 MOV #SSWREG,SWR ;;NO,USE APT SWITCH REGISTER
1312 002756 67$:
1313
1314 ; SET UP FPP UNEXPECTED TRAP CATCHER - - - - -
1315 002756 012737 023316 000244 MOV #FPPILT,#FPPVEC ; NEW PC AT FPP TRAP
1316 002764 005037 000246 CLR #FPPVEC+2 ; NEW PS AT FPP TRAP

```

```

1317
1318 002770 104401 002376          TYPE      ,BGNMES          ; ID MESSAGE AT START
1319
1320 ;////////////////////////////////////
1321 ; MESSAGE ON WHETHER OR NOT HFP UNIT IS PRESENT
1322
1323 002774 076600 000022          MED      RWHAMI          ; WHAMI INTO RO
1324 003000 032700 000020          BIT      #BIT04,RO      ; IS THERE A HFP UNIT ?
1325 003004 001403                    BEQ      70$            ; NO, BR
1326 003006 104401 003022          TYPE      68$          ; INDICATE FP11-E PRESENT
1327 003012 000453                    BR      NEWPAS         ; GO FOR SUBPASS INIT
1328 003014 104401 003062          70$:      TYPE      69$          ; INDICATE NO FP11-E
1329 003020 000450                    BR      NEWPAS         ; GO FOR SUBPASS INIT
1330
1331 003022 005015 020052 050106 68$:      .ASCIZ  <15><12>*" FP11-E HFP UNIT PRESENT *"<15><12>
1332 003030 030461 042455 044040
1333 003036 050106 052440 044516
1334 003044 020124 051120 051505
1335 003052 047105 020124 006452
1336 003060 000012
1337 003062 005015 020052 047516 69$:      .ASCIZ  <15><12>*" NO FP11-E HFP UNIT - ALL TESTS WFP ONLY *"<15><12>
1338 003070 043040 030520 026461
1339 003076 020105 043110 020120
1340 003104 047125 052111 026440
1341 003112 040440 046114 052040
1342 003120 051505 051524 053440
1343 003126 050106 047440 046116
1344 003134 020131 006452 000012
1345          .EVEN
1346
1347 ;////////////////////////////////////
1348
1349 ;*****
1350 ; NEW PASS ENTERS HERE
1351 ;*****
1352
1353
1354 003142 012706 001100          NEWPAS: MOV      #STACK,SP          ; RESET STACK PTR
1355
1356 003146 032777 010000 175772          BIT      #BIT12,JSWR      ; INHIBIT STATUS TYPEOUTS ?
1357 003154 001011                    BNE     SUBPAS         ; BR IF YES
1358
1359 003156 104401 002475          TYPE      NWPAS1        ; "PASS #"
1360 003162 013746 001332          MOV      $PASS,-(SP)     ; PASS COUNT INTO ...
1361 003166 005216                    INC     (SP)             ; 1-N RANGE
1362 003170 104403                    TYPOS   ; TYPE OCTAL
1363 003172      006      000          .BYTE   6,0             ; 6 DIGITS, NO LEADING ZEROS
1364 003174 104401 001321          TYPE     , $CRLF        ; END THE LINE
1365
1366
1367 ;*****
1368 ; NEW SUBPASS ENTERS HERE
1369 ;*****
1370
1371 003200 076600 000022          SUBPAS: MED      RWHAMI          ; GET WHAMI INTO RO
1372 003204 032700 000020          BIT      #BIT04,RO      ; 1=HFP PRESENT, 0=NO

```

1373	003210	001430				BEQ	20\$													
1374																				
1375	003212	076600	000144			MED	,RFLAG													
1376																				
1377	003216	032777	000002	175722		BIT	#SW01, @SWR													
1378	003224	001413				BEQ	1\$													
1379																				
1380	003226	032777	000001	175712		BIT	#SW00, @SWR													
1381	003234	001403				BEQ	2\$													
1382	003236	042700	010000			BIC	#BIT12, RO													
1383	003242	000402				BR	3\$													
1384	003244	052700	010000		2\$:	BIS	#BIT12, RO													
1385	003250	076600	000344		3\$:	MED	,WFLAG													
1386																				
1387	003254	032700	010000		1\$:	BIT	#BIT12, RO													
1388	003260	001404				BEQ	20\$													
1389																				
1390	003262	012737	025526	024100	19\$:	MOV	#ASCHOT, HOTWRM													
1391	003270	000403				BR	21\$													
1392	003272	012737	025534	024100	20\$:	MOV	#ASCWRM, HOTWRM													
1393	003300	005037	001102		21\$:	CLR	\$TSTNM													

```

;IF NO HFP, TEST WARM ONLY
;GET FLAGS INTO RO
;SW01: 1=HFP OR WFP TEST ONLY
;      0=ALTERNATE HFP/WFP PER PASS
;SW00: 1=WFP ONLY
;      0=HFP ONLY
;CLEAR HFP ENABLE FLAG<5> FOR WFP
;SET HFP ENABLE FLAG<5> FOR HFP
;REWRITE FLAGS
;TEST WHO'S ENABLED: HOT, WARM
;SET APPROPRIATE HEADER:
;"HOT: "
;"WARM: "
;ALL DONE, RESET TEST NUMBER COUNTER
    
```

# E03

FPU BASIC INSTR TESTS MACY11 27(1006) 09-FEB-77 09:48 PAGE 32  
 DQFPAA.P11 09-FEB-77 09:42 T1 TEST OF FPS REGISTER BY RIPPLING A 1

```

1394
1395
1396
1397 003304 000004
1398 003306 012700 000001
1399 003312 012737 003320 001114
1400
1401 003320 010037 001170
1402 003324 042737 000020 001170
1403 003332 170137 001170
1404 003336 042737 030000 001170
1405 003344 170237 001172
1406 003350 023737 001170 001172
1407 003356 001401
1408 003360 104001
1409 003362 000241
1410 003364 006100
1411 003366 103354
1412
1413
1414
1415
1416
1417
1418
1419 003370 000004
1420 003372 012700 177776
1421 003376 012737 003404 001114
1422
1423 003404 010037 001170
1424 003410 042737 000020 001170
1425 003416 170137 001170
1426 003422 042737 030000 001170
1427 003430 170237 001172
1428 003434 023737 001170 001172
1429 003442 001401
1430 003444 104002
1431 003446 000261
1432 003450 006100
1433 003452 103754

;*****
;#TEST 1 TEST OF FPS REGISTER BY RIPPLING A 1
;*****
↑ST1: SCOPE
MOV #BIT00,RO ; INITIAL PATTERN
MOV #1$,SLPERR ; ERROR LOOPING RETURN
1$: MOV RO,$REGO ; EDIT PATTERN SO BIT 4 OF FPS ( FMM )
BIC #BIT04,$REGO ; WILL ALWAYS BE OFF DURING TEST
LDFPS $REGO ; LOAD FPS
BIC #BIT13!BIT12,$REGO ; BITS 13 AND 12 ALWAYS READ 0
STFPS $REG1 ; STORE FPS
CMP $REGO,$REG1 ; LOAD/STORE WORK ?
BEQ 2$ ; YES
ERROR 1 ; NO - ERROR RETURN
2$: CLC ; ROTATE IN A 0
ROL RO ; SHIFT PATTERN 1 POSITION LEFT
BCC 1$ ; CONT IF NOT YET FINISHED

;*****
;#TEST 2 TEST OF FPS REGISTER BY RIPPLING A 0
;*****
↑ST2: SCOPE
MOV #1CBIT00,RO ; INITIAL PATTERN
MOV #1$,SLPERR ; ERROR LOOPING RETURN
1$: MOV RO,$REGO ; EDIT PATTERN SO BIT 4 OF FPS ( FMM )
BIC #BIT04,$REGO ; WILL ALWAYS BE OFF DURING TEST
LDFPS $REGO ; LOAD FPS
BIC #BIT13!BIT12,$REGO ; BITS 13 AND 12 ALWAYS READ 0
STFPS $REG1 ; STORE FPS
CMP $REGO,$REG1 ; LOAD/STORE WORK ?
BEQ 2$ ; YES
ERROR 2 ; NO - ERROR RETURN
2$: SEC ; ROTATE IN A 1
ROL RO ; SHIFT PATTERN 1 POSITION LEFT
BCS 1$ ; CONT IF NOT YET FINISHED
  
```

```

1434
1435
1436
1437 003454 000004
1438 003456 170127 147757
1439 003462 170001
1440 003464 170237 001170
1441 003470 022737 147557 001170
1442 003476 001401
1443 003500 104003
1444 003502
1445
1446
1447
1448
1449
1450
1451
1452
1453 003502 000004
1454 003504 170127 000000
1455 003510 170011
1456 003512 170237 001170
1457 003516 022737 000200 001170
1458 003524 001401
1459 003526 104004
1460 003530
1461
1462
1463
1464
1465
1466
1467
1468
1469 003530 000004
1470 003532 170127 147757
1471 003536 170002
1472 003540 170237 001170
1473 003544 022737 147657 001170
1474 003552 001401
1475 003554 104005
1476 003556
1477
1478
1479
1480
1481
1482
1483
1484
1485 003556 000004
1486 003560 170127 000000
1487 003564 170012
1488 003566 170237 001170
1489 003572 022737 000100 001170

```

```

*****
*TEST 3 TEST OF SETF INSTRUCTION
*****
TST3: SCOPE
LDFPS #FPS1 ; INITIAL PATTERN
SETF ; TEST IT
STFPS $REGO ; GET NEW FPS
CMP #FPS1&↑CBIT07,$REGO ; DID SETF CLEAR FPS BIT 7 ?
BEQ 1$
ERROR 3 ; NO - ERROR
; YES - NEXT TEST
1$:

```

```

*****
*TEST 4 TEST OF SETD INSTRUCTION
*****
TST4: SCOPE
LDFPS #FPS0 ; INITIAL PATTERN
SETD ; TEST IT
STFPS $REGO ; GET NEW FPS
CMP #FPS0!BIT07,$REGO ; DID SETD SET FPS BIT 7 ?
BEQ 1$
ERROR 4 ; NO - ERROR
; YES - NEXT TEST
1$:

```

```

*****
*TEST 5 TEST OF SETI INSTRUCTION
*****
TST5: SCOPE
LDFPS #FPS1 ; INITIAL PATTERN
SETI ; TEST IT
STFPS $REGO ; GET NEW FPS
CMP #FPS1&↑CBIT06,$REGO ; DID SETI CLEAR FPS BIT 6 ?
BEQ 1$
ERROR 5 ; NO - ERROR
; YES - NEXT TEST
1$:

```

```

*****
*TEST 6 TEST OF SETL INSTRUCTION
*****
TST6: SCOPE
LDFPS #FPS0 ; INITIAL PATTERN
SETL ; TEST IT
STFPS $REGO ; GET NEW FPS
CMP #FPS0!BIT06,$REGO ; DID SETL SET FPS BIT 6 ?

```



G03

FPU BASIC INSTR TESTS MACY11 27(1006) 09-FEB-77 09:48 PAGE 34  
DQFPAA.P11 09-FEB-77 09:42 T6 TEST OF SETL INSTRUCTION

1490 003600 001401  
1491 003602 104006  
1492 003604  
1493

1\$:  
BEQ 1\$  
ERROR 6

:  
: NO - ERROR  
: YES - NEXT TEST

• 8 9

1494  
1495  
1496  
1497  
1498  
1499  
1500  
1501  
1502  
1503  
1504  
1505  
1506  
1507  
1508  
1509  
1510  
1511  
1512  
1513  
1514  
1515  
1516  
1517  
1518  
1519  
1520  
1521  
1522  
1523  
1524  
1525  
1526  
1527  
1528  
1529  
1530  
1531  
1532  
1533  
1534  
1535  
1536  
1537  
1538  
1539  
1540  
1541  
1542  
1543  
1544  
1545  
1546  
1547  
1548  
1549

```

;*****
; .DSABL AMA ; ASSEMBLE ALL REFERENCES AS THEY ARE PRINTED
;*****
;*****
; *TEST 7 TEST OF LOAD-SRC MODE-0, STORE-DST MODE-7(PC)
;*****
TST7: SCOPE
MOV ADDR1,$REG0 ; GET TEST PATTERN
MOV $REG0,R5 ; DATA
LDFPS R5 ; MO-R5
MOV #SREG1,$REG2 ; ADDR(DEST)
STFPS @SREG2 ; M7-R7
CMP $REG0,$REG1 ; LOAD/STORE WORK?
BEQ 64$ ;
ERROR 7 ; NOT EQUAL, SIGNAL ERROR
64$:
BR TST10 ;;
ADDR1: .WORD 105252 ; TEST PATTERN
;*****
; *TEST 10 TEST OF LOAD-SRC MODE-1, STORE-DST MODE-3
;*****
TST10: SCOPE
MOV ADDR2,$REG0 ; GET TEST PATTERN
MOV #SREG0,R3 ; ADDR(DATA)
LDFPS (R3) ; M1-R3
MOV #SREG1,$REG2 ; ADDR(DEST)
MOV #SREG2,R5 ; ADDR(ADDR(DEST))
STFPS @R5+ ; M3-R5
CMP R5,#SREG2+2 ; WAS DST ADDR REG INCRE RIGHT AMOUNT?
BEQ 64$ ;
ERROR 10 ; NOT EQUAL, SIGNAL ERROR
64$:
CMP $REG0,$REG1 ; LOAD/STORE WORK?
BEQ 65$ ;
ERROR 11 ; NOT EQUAL, SIGNAL ERROR
65$:
BR TST11 ;;
ADDR2: .WORD 042505 ; TEST PATTERN
;*****

```

```

1550 ;*TEST 11 TEST OF LOAD-SRC MODE-2, STORE-DST MODE-5
1551 ;*****
1552 003732 000004 TST11: SCOPE
1553 003734 016767 000060 175226 MOV ADDR3,$REG0 ; GET TEST PATTERN
1554
1555 003742 012701 001170 MOV #$REG0,R1 ; ADDR(DATA)
1556 003746 170121 LDFPS (R1)+ ; M2-R1
1557
1558 003750 012767 001172 175216 MOV #$REG1,$REG2 ; ADDR(DEST)
1559 003756 012703 001176 MOV #$REG2+2,R3 ; ADDR(ADDR(DEST)+2)
1560 003762 170253 STFPS 2-(R3) ; M5-R3
1561
1562 003764 020127 001172 CMP R1,$REG0+2 ; WAS SRC ADDR REG INCRE RIGHT AMOUNT?
1563 003770 001401 BEQ 64$ ;
1564 003772 104012 ERROR 12 ; NOT EQUAL, SIGNAL ERROR
1565 003774 64$:
1566
1567 003774 020327 001174 CMP R3,$REG2 ; WAS DST ADDR REG DECRE RIGHT AMOUNT?
1568 004000 001401 BEQ 65$ ;
1569 004002 104013 ERROR 13 ; NOT EQUAL, SIGNAL ERROR
1570 004004 65$:
1571
1572 004004 026767 175160 175160 CMP $REG0,$REG1 ; LOAD/STORE WORK?
1573 004012 001401 BEQ 66$ ;
1574 004014 104014 ERROR 14 ; NOT EQUAL, SIGNAL ERROR
1575 004016 66$:
1576
1577 004016 000401 BR TST12 ;;
1578
1579 004020 105252 ADDR3: .WORD 105252 ; TEST PATTERN
1580
1581 ;*****
1582 ;*TEST 12 TEST OF LOAD-SRC MODE-3, STORE-DST MODE-6(PC)
1583 ;*****
1584
1585 004022 000004 TST12: SCOPE
1586 004024 016767 000046 175136 MOV ADDR4,$REG0 ; GET TEST PATTERN
1587
1588 004032 012767 001170 175134 MOV #$REG0,$REG2 ; ADDR(DATA)
1589 004040 012700 001174 MOV #$REG2,R0 ; ADDR(ADDR(DATA))
1590 004044 170130 LDFPS 2(R0)+ ; M3-R0
1591
1592 004046 170267 175120 STFPS $REG1 ; M6-R7
1593
1594 004052 020027 001176 CMP R0,$REG2+2 ; WAS SRC ADDR REG INCRE RIGHT AMOUNT?
1595 004056 001401 BEQ 64$ ;
1596 004060 104015 ERROR 15 ; NOT EQUAL, SIGNAL ERROR
1597 004062 64$:
1598
1599 004062 026767 175102 175102 CMP $REG0,$REG1 ; LOAD/STORE WORK?
1600 004070 001401 BEQ 65$ ;
1601 004072 104016 ERROR 16 ; NOT EQUAL, SIGNAL ERROR
1602 004074 65$:
1603
1604 004074 000401 BR TST13 ;;
1605

```

```

1606 004076 042505 ADDR4: .WORD 042505 ; TEST PATTERN
1607
1608
1609
1610
1611
1612 004100 000004
1613 004102 016767 000042 175060
1614
1615 004110 012702 001172
1616 004114 170142
1617
1618 004116 170200
1619 004120 010067 175046
1620
1621 004124 020227 001170
1622 004130 001401
1623 004132 104017
1624 004134
1625
1626 004134 026767 175030 175030
1627 004142 001401
1628 004144 104020
1629 004146
1630
1631 004146 000401
1632
1633 004150 105252 ADDR5: .WORD 105252 ; TEST PATTERN
1634
1635
1636
1637
1638
1639 004152 000004
1640 004154 016767 000046 175006
1641
1642 004162 012767 001170 175004
1643 004170 012704 001176
1644 004174 170154
1645
1646 004176 170237 001172
1647
1648 004202 020427 001174
1649 004206 001401
1650 004210 104021
1651 004212
1652
1653 004212 026767 174752 174752
1654 004220 001401
1655 004222 104022
1656 004224
1657
1658 004224 000401
1659
1660 004226 042505 ADDR6: .WORD 042505 ; TEST PATTERN
1661

```

```

*****
; TEST 13 TEST OF LOAD-SRC MODE-4, STORE-DST MODE-0
*****
TST13: SCOPE
MOV ADDR5,$REG0 ; GET TEST PATTERN
MOV #$REG0+2,R2 ; ADDR(DATA+2)
LDFPS -(R2) ; M4-R2
STFPS R0 ; M0-R0
MOV R0,$REG1 ; DEST
CMP R2,$REG0 ; WAS SRC ADDR REG DECRE RIGHT AMOUNT?
BEQ 64$
ERROR 17 ; NOT EQUAL, SIGNAL ERROR
64$:
CMP $REG0,$REG1 ; LOAD/STORE WORK?
BEQ 65$
ERROR 20 ; NOT EQUAL, SIGNAL ERROR
65$:
BR TST14 ;;

```

```

*****
; TEST 14 TEST OF LOAD-SRC MODE-5, STORE-DST MODE-3(PC)
*****
TST14: SCOPE
MOV ADDR6,$REG0 ; GET TEST PATTERN
MOV #$REG0,$REG2 ; ADDR(DATA)
MOV #$REG2+2,R4 ; ADDR(ADDR(DATA)+2)
LDFPS 2-(R4) ; M5-R4
STFPS 2#$REG1 ; M3-R7
CMP R4,$REG2 ; WAS SRC ADDR REG DECRE RIGHT AMOUNT?
BEQ 64$
ERROR 21 ; NOT EQUAL, SIGNAL ERROR
64$:
CMP $REG0,$REG1 ; LOAD/STORE WORK?
BEQ 65$
ERROR 22 ; NOT EQUAL, SIGNAL ERROR
65$:
BR TST15 ;;

```

K03

1662  
1663  
1664  
1665  
1666  
1667  
1668  
1669  
1670  
1671  
1672  
1673  
1674  
1675  
1676  
1677  
1678  
1679  
1680  
1681  
1682  
1683  
1684  
1685  
1686  
1687  
1688  
1689  
1690  
1691  
1692  
1693  
1694  
1695  
1696  
1697  
1698  
1699  
1700  
1701  
1702  
1703  
1704  
1705  
1706  
1707  
1708  
1709  
1710  
1711  
1712  
1713  
1714  
1715  
1716  
1717

\*\*\*\*\*  
; \*TEST 15 TEST OF LOAD-SRC MODE-6, STORE-DST MODE-7  
; \*\*\*\*\*

```
TST15: SCOPE
MOV ADDR7,$REG0 ; GET TEST PATTERN
MOV #$REG0+64,R3 ; ADDR(DATA)+64
LDFPS -64(R3) ; M6-R3
MOV #$REG1,$REG2 ; ADDR(DEST)
MOV #$REG2-40,R1 ; ADDR(ADDR(DEST))-40
STFPS 240(R1) ; M7-R1
CMP $REG0,$REG1 ; LOAD/STORE WORK?
BEQ 64$ ;
ERROR 14 ; NOT EQUAL, SIGNAL ERROR
64$:
BR TST16 ;;
ADDR7: .WORD 105252 ; TEST PATTERN
```

\*\*\*\*\*  
; \*TEST 16 TEST OF LOAD-SRC MODE-7, STORE-DST MODE-2(PC)  
; \*\*\*\*\*

```
TST16: SCOPE
MOV ADDR10,$REG0 ; GET TEST PATTERN
MOV #$REG0,$REG2 ; ADDR(DATA)
MOV #$REG2+22,R2 ; ADDR(ADDR(DATA))+22
LDFPS 2-22(R2) ; M7-R2
MOV #ERROR117,1$ ; SETUP ERROR CALL FOR WRONG INCREMENT
STFPS (PC)+ ; M2-R7
1$: .WORD 0 ; DEST
BR 2$ ; THIS SHOULD BE NEXT INSTR AFTER FPP PC MODE 2
ERROR 120 ; NOT HERE (+4 INCRE)
ERROR 121 ; OR HERE (+6 INCRE)
ERROR 122 ; OR HERE (+10 INCRE)
2$: MOV 1$, $REG1 ; GET DEST
CMP $REG0,$REG1 ; LOAD/STORE WORK?
BEQ 64$ ;
ERROR 23 ; NOT EQUAL, SIGNAL ERROR
64$:
BR TST17 ;;
ADDR10: .WORD 042505 ; TEST PATTERN
```

\*\*\*\*\*  
; \*TEST 17 TEST OF LOAD-SRC MODE-2(PC), STORE-DST MODE-4  
; \*\*\*\*\*

```

1718 004400 000004 TST17: SCOPE
1719 004402 016767 000056 174560 MOV ADDR11,$REG0 ; GET TEST PATTERN
1720
1721 004410 016767 174554 000002 MOV $REG0,1$ ; PUT DATA
1722 004416 170127 LDFPS (PC)+ ; M2-R7
1723 004420 000000 1$: .WORD 0 ; DATA
1724 004422 000403 BR 2$ ; THIS SHOULD BE NEXT INSTR AFTER FPP PC MODE 2
1725 004424 104120 ERROR 120 ; NOT HERE (+4 INCRE)
1726 004426 104121 ERROR 121 ; OR HERE (+6 INCRE)
1727 004430 104122 ERROR 122 ; OR HERE (+10 INCRE)
1728
1729 004432 012703 001174 2$: MOV #$REG1+2,R3 ; ADDR(DEST+2)
1730 004436 170243 STFPS -(R3) ; M4-R3
1731
1732 004440 020327 001172 CMP R3,$REG1 ; WAS DST ADDR REG DECRE RIGHT AMOUNT?
1733 004444 001401 BEQ 64$ ;
1734 004446 104024 ERROR 24 ; NOT EQUAL, SIGNAL ERROR
1735 004450 64$:
1736
1737 004450 026767 174514 174514 CMP $REG0,$REG1 ; LOAD/STORE WORK?
1738 004456 001401 BEQ 65$ ;
1739 004460 104025 ERROR 25 ; NOT EQUAL, SIGNAL ERROR
1740 004462 65$:
1741
1742 004462 000401 BR TST20 ;;
1743
1744 004464 104117 ADDR11: .WORD ERROR117 ; TEST PATTERN
1745
1746
1747
1748
1749

```

```

*****
*TEST 20 TEST OF LOAD-SRC MODE-3(PC), STORE-DST MODE-6
*****

```

```

1750 004466 000004 TST20: SCOPE
1751 004470 016767 000032 174472 MOV ADDR12,$REG0 ; GET TEST PATTERN
1752
1753 004476 170137 001170 LDFPS 2*$REG0 ; M3-R7
1754
1755 004502 012702 001144 MOV #$REG1-26,R2 ; ADDR(DEST)-26
1756 004506 170262 000026 STFPS 26(R2) ; M6-R2
1757
1758 004512 026767 174452 174452 CMP $REG0,$REG1 ; LOAD/STORE WORK?
1759 004520 001401 BEQ 64$ ;
1760 004522 104026 ERROR 26 ; NOT EQUAL, SIGNAL ERROR
1761 004524 64$:
1762
1763 004524 000401 BR TST21 ;;
1764
1765 004526 042505 ADDR12: .WORD 042505 ; TEST PATTERN
1766
1767

```

```

*****
*TEST 21 TEST OF LOAD-SRC MODE-6(PC), STORE-DST MODE-1
*****

```

```

1770
1771 004530 000004 TST21: SCOPE
1772 004532 016767 000030 174430 MOV ADDR13,$REG0 ; GET TEST PATTERN
1773

```

M03

FPU BASIC INSTR TESTS MACY11 27(1006) 09-FEB-77 09:48 PAGE 40  
 DGFPA.P11 09-FEB-77 09:42 T21 TEST OF LOAD-SRC MODE-6(PC), STORE-DST MODE-1

```

1774 004540 170167 174424          LDFPS  $REG0          ; M6-R7
1775
1776 004544 012702 001172          MOV    #SREG1,R2     ; ADDR(DEST)
1777 004550 170212                    STFPS  (R2)          ; M1-R2
1778
1779 004552 026767 174412 174412    CMP    $REG0,$REG1   ; LOAD/STORE WORK?
1780 004560 001401                    BEQ    64$           ;
1781 004562 104026                    ERROR  26           ; NOT EQUAL, SIGNAL ERROR
1782 004564                    64$:
1783
1784 004564 000401                    BR     TST22        ;;
1785
1786 004566 105252          ADDR13: .WORD 105252 ; TEST PATTERN
1787
1788
1789
1790
1791
1792 004570 000004          ;*****
1793 004572 016767 000046 174370    ;*TEST 22 TEST OF LOAD-SRC MODE-7(PC), STORE-DST MODE-2
1794
1795 004600 012767 001170 174366    ;*****
1796 004606 170177 174362          †ST22: SCOPE
1797
1798 004612 012704 001172          MOV    ADDR14,$REG0 ; GET TEST PATTERN
1799 004616 170224                    MOV    #SREG0,$REG2 ; ADDR(DATA)
1800
1801 004620 020427 001174          LDFPS  3$REG2       ; M7-R7
1802 004624 001401                    MOV    #SREG1,R4    ; ADDR(DEST)
1803 004626 104027                    STFPS  (R4)+        ; M2-R4
1804 004630                    64$:
1805
1806 004630 026767 174334 174334    CMP    R4,#SREG1+2  ; WAS DST ADDR REG INCRE RIGHT AMOUNT?
1807 004636 001401                    BEQ    64$           ;
1808 004640 104022                    ERROR  27           ; NOT EQUAL, SIGNAL ERROR
1809 004642                    65$:
1810
1811 004642 000401                    BR     TST23        ;;
1812
1813 004644 042505          ADDR14: .WORD 042505 ; TEST PATTERN
1814
1815
  
```

```

1816
1817
1818 ;:*****
1819 .ENABL AMA ; BEGIN ASSEMBLING RELATIVE REFERENCES AS ABSOLUTE
1820 ;:*****
1821 ;:*****
1822 ;:TEST 23 TEST OF CFCC INSTRUCTION
1823 ;:*****
1824 TST23: SCOPE
1825 004646 000004 MOV #10,R0 ; NUMBER OF ENTRIES IN TABLE
1826 004650 012700 000010 MOV #OFCC,R1 ; START OF TABLE
1827 004654 012701 004760 MOV #15,$LPERR ; ERROR LOOPING RETURN
1828 004660 012737 004666 001114
1829 004666 011137 001170 15: MOV (R1),SREG0 ; GET FIRST FCC PATTERN
1830 004672 042737 177760 001170 BIC #CCONLY,SREG0 ; BITS IN FCC POSITIONS ONLY
1831 004700 170137 001170 LDFPS SREG0 ; STORE IN FPS REGISTER
1832 004704 032711 000040 BIT #BIT05,(R1) ; TEST WHETHER FLOATING A 1 OR 0
1833 004710 001002 BNE 25
1834 004712 000257 CCC ; FOR FLOAT A 1, START W/ CC = 0000
1835 004714 000401 BR 35
1836 004716 000277 25: SCC ; FOR FLOAT A 0, START W/ CC = 1111
1837 004720 170000 35: CFCC ; COPY THE CONDITION CODES
1838 004722 013737 177776 001172 MOV #PSW,SREG1 ; GET CPU CC BITS
1839 004730 042737 177760 001172 BIC #CCONLY,SREG1 ; CLEAR EXTRANEIOUS BITS
1840 004736 023737 001170 001172 CMP SREG0,SREG1 ; WERE THEY COPIED OK ?
1841 004744 001401 BEQ 45
1842 004746 104030 ERROR 30 ; NO - SIGNAL ERROR
1843 004750 062701 000002 45: ADD #2,R1 ; INCRE R1 OUT OF ERROR LOOP
1844 004754 077034 SOB R0,15 ; LOOP CONTROL
1845 004756 000410 BR TST24 ;:
1846
1847 004760 000001 000002 000004 OFCC: .WORD 000001,000002,000004,000010 ; TABLE OF CC
1848 004766 000010
1849 004770 000056 000055 000053 .WORD 000056,000055,000053,000047 ; TEST PATTERNS
1850 004776 000047

```



1851  
1852  
1853  
1854 005000 000004  
1855 005002 012704 020706  
1856 005006 012705 005020  
1857 005012 004737 020524  
1858  
1859 005016 000413  
1860  
1861 005020  
1862 005020 000000 000000 000000  
1863 005026 000000  
1864 005030 000000 000000 000000  
1865 005036 000000  
1866 005040 047653 047644  
1867 005044 000000  
1868  
1869  
1870  
1871  
1872 005046 000004  
1873 005050 012704 020706  
1874 005054 012705 005066  
1875 005060 004737 020524  
1876  
1877 005064 000413  
1878  
1879 005066  
1880 005066 177777 177777 177777  
1881 005074 177777  
1882 005076 177777 177777 177777  
1883 005104 177777  
1884 005106 047745 047750  
1885 005112 000000  
1886  
1887  
1888  
1889  
1890 005114 000004  
1891 005116 012704 020706  
1892 005122 012705 005134  
1893 005126 004737 020524  
1894  
1895 005132 000413  
1896  
1897 005134  
1898 005134 052525 052525 052525  
1899 005142 052525  
1900 005144 052525 052525 052525  
1901 005152 052525  
1902 005154 047652 047640  
1903 005160 000000  
1904  
1905  
1906

```
*****  
; *TEST 24 TEST FP ACO, D MODE, TEST PATTERN LDAD-1  
*****  
TST24: SCOPE  
MOV #LDARDO,R4 ; PTR TO TESTING ROUTINE  
MOV #LDAD1,R5 ; PTR TO TEST DATA SET  
JSR PC,@#LDADT ; GO TEST  
  
BR TST25 ;;  
  
LDAD1: ; TEST DATA SET LDAD-1:  
.WORD 0, 0, 0, 0 ; PATTERN 1  
  
.WORD 0, 0, 0, 0 ; PATTERN EXPECTED AFTER  
  
.WORD 047653,047644 ; FPS: BEFORE, AFTER  
.WORD 000000 ; FEC AFTER ( 0 = N/A )  
  
*****  
; *TEST 25 TEST FP ACO, D MODE, TEST PATTERN LDAD-2  
*****  
TST25: SCOPE  
MOV #LDARDO,R4 ; PTR TO TESTING ROUTINE  
MOV #LDAD2,R5 ; PTR TO TEST DATA SET  
JSR PC,@#LDADT ; GO TEST  
  
BR TST26 ;;  
  
LDAD2: ; TEST DATA SET LDAD-2:  
.WORD M1, M1, M1, M1 ; PATTERN 2  
  
.WORD M1, M1, M1, M1 ; PATTERN EXPECTED AFTER  
  
.WORD 047745,047750 ; FPS: BEFORE, AFTER  
.WORD 000000 ; FEC AFTER ( 0 = N/A )  
  
*****  
; *TEST 26 TEST FP ACO, D MODE, TEST PATTERN LDAD-3  
*****  
TST26: SCOPE  
MOV #LDARDO,R4 ; PTR TO TESTING ROUTINE  
MOV #LDAD3,R5 ; PTR TO TEST DATA SET  
JSR PC,@#LDADT ; GO TEST  
  
BR TST27 ;;  
  
LDAD3: ; TEST DATA SET LDAD-3:  
.WORD ALTP,ALTP,ALTP,ALTP ; PATTERN 3  
  
.WORD ALTP,ALTP,ALTP,ALTP ; PATTERN EXPECTED AFTER  
  
.WORD 047652,047640 ; FPS: BEFORE, AFTER  
.WORD 000000 ; FEC AFTER ( 0 = N/A )  
  
*****  
; *TEST 27 TEST FP ACO, D MODE, TEST PATTERN LDAD-4
```

```

1907
1908 005162 000004
1909 005164 012704 020706
1910 005170 012705 005202
1911 005174 004737 020524
1912
1913 005200 000413
1914
1915 005202
1916 005202 125252 125252 125252
1917 005210 125252
1918 005212 125252 125252 125252
1919 005220 125252
1920 005222 047704 047710
1921 005226 000000
1922
1923
1924
1925
1926 005230 000004
1927 005232 012704 020706
1928 005236 012705 005250
1929 005242 004737 020524
1930
1931 005246 000413
1932
1933 005250
1934 005250 100177 177777 177777
1935 005256 177777
1936 005260 100177 177777 177777
1937 005266 177777
1938 005270 043743 043754
1939 005274 000000
1940
1941
1942
1943
1944 005276 000004
1945 005300 012704 020706
1946 005304 012705 005316
1947 005310 004737 020524
1948
1949 005314 000413
1950
1951 005316
1952 005316 100000 000000 000000
1953 005324 000000
1954 005326 052525 177777 125252
1955 005334 000000
1956 005336 047602 147614
1957 005342 100014
1958
1959
1960
1961
1962 005344 000004

```

```

*****
TST27: SCOPE
MOV #LDARDO,R4 ; PTR TO TESTING ROUTINE
MOV #LDAD4,R5 ; PTR TO TEST DATA SET
JSR PC,@#LDADT ; GO TEST
BR TST30 ;;

LDAD4: ; TEST DATA SET LDAD-4:
.WORD ALTN,ALTN,ALTN,ALTN ; PATTERN 4
.WORD ALTN,ALTN,ALTN,ALTN ; PATTERN EXPECTED AFTER
.WORD 047704,047710 ; FPS: BEFORE, AFTER
.WORD 000000 ; FEC AFTER ( 0 = N/A )

*****
*TEST 30 TEST FP ACO, D MODE, TEST PATTERN LDAD-5
*****
TST30: SCOPE
MOV #LDARDO,R4 ; PTR TO TESTING ROUTINE
MOV #LDAD5,R5 ; PTR TO TEST DATA SET
JSR PC,@#LDADT ; GO TEST
BR TST31 ;;

LDAD5: ; TEST DATA SET LDAD-5:
.WORD ZX1MN,M1,M1,M1 ; PATTERN 5
.WORD ZX1MN,M1,M1,M1 ; PATTERN EXPECTED AFTER
.WORD 043743,043754 ; FPS: BEFORE, AFTER
.WORD 000000 ; FEC AFTER ( 0 = N/A )

*****
*TEST 31 TEST FP ACO, D MODE, TEST PATTERN LDAD-6
*****
TST31: SCOPE
MOV #LDARDO,R4 ; PTR TO TESTING ROUTINE
MOV #LDAD6,R5 ; PTR TO TEST DATA SET
JSR PC,@#LDADT ; GO TEST
BR TST32 ;;

LDAD6: ; TEST DATA SET LDAD-6:
.WORD MO, 0, 0, 0 ; PATTERN 6
.WORD ALTP,M1, ALTN,0 ; PATTERN EXPECTED AFTER
.WORD 047602,147614 ; FPS: BEFORE, AFTER
.WORD 100014 ; FEC AFTER ( 0 = N/A )

*****
*TEST 32 TEST FP AC1, D MODE, TEST PATTERN LDAD-1
*****
TST32: SCOPE

```

1963 005346 012704 020722  
1964 005352 012705 005020  
1965 005356 004737 020524  
1966  
1967 005362 000400  
1968  
1969  
1970  
1971  
1972  
1973  
1974 005364 000004  
1975 005366 012704 020722  
1976 005372 012705 005066  
1977 005376 004737 020524  
1978  
1979 005402 000400  
1980  
1981  
1982  
1983  
1984  
1985  
1986 005404 000004  
1987 005406 012704 020722  
1988 005412 012705 005134  
1989 005416 004737 020524  
1990  
1991 005422 000400  
1992  
1993  
1994  
1995  
1996  
1997  
1998 005424 000004  
1999 005426 012704 020722  
2000 005432 012705 005202  
2001 005436 004737 020524  
2002  
2003 005442 000400  
2004  
2005  
2006  
2007  
2008  
2009  
2010 005444 000004  
2011 005446 012704 020722  
2012 005452 012705 005250  
2013 005456 004737 020524  
2014  
2015 005462 000400  
2016  
2017  
2018

```
MOV #LDAD1,R4 ; PTR TO TESTING ROUTINE
MOV #LDAD1,R5 ; PTR TO TEST DATA SET
JSR PC,@#LDADT ; GO TEST

BR TST33 ;;

; ( FOR DATA SEE TEST OF FP REG ACO WITH TEST PATTERN LDAD-1, ABOVE )

;*****
;#TEST 33 TEST FP AC1, D MODE, TEST PATTERN LDAD-2
;*****
TST33: SCOPE
MOV #LDAD1,R4 ; PTR TO TESTING ROUTINE
MOV #LDAD2,R5 ; PTR TO TEST DATA SET
JSR PC,@#LDADT ; GO TEST

BR TST34 ;;

; ( FOR DATA SEE TEST OF FP REG ACO WITH TEST PATTERN LDAD-2, ABOVE )

;*****
;#TEST 34 TEST FP AC1, D MODE, TEST PATTERN LDAD-3
;*****
TST34: SCOPE
MOV #LDAD1,R4 ; PTR TO TESTING ROUTINE
MOV #LDAD3,R5 ; PTR TO TEST DATA SET
JSR PC,@#LDADT ; GO TEST

BR TST35 ;;

; ( FOR DATA SEE TEST OF FP REG ACO WITH TEST PATTERN LDAD-3, ABOVE )

;*****
;#TEST 35 TEST FP AC1, D MODE, TEST PATTERN LDAD-4
;*****
TST35: SCOPE
MOV #LDAD1,R4 ; PTR TO TESTING ROUTINE
MOV #LDAD4,R5 ; PTR TO TEST DATA SET
JSR PC,@#LDADT ; GO TEST

BR TST36 ;;

; ( FOR DATA SEE TEST OF FP REG ACO WITH TEST PATTERN LDAD-4, ABOVE )

;*****
;#TEST 36 TEST FP AC1, D MODE, TEST PATTERN LDAD-5
;*****
TST36: SCOPE
MOV #LDAD1,R4 ; PTR TO TESTING ROUTINE
MOV #LDAD5,R5 ; PTR TO TEST DATA SET
JSR PC,@#LDADT ; GO TEST

BR TST37 ;;

; ( FOR DATA SEE TEST OF FP REG ACO WITH TEST PATTERN LDAD-5, ABOVE )
```

E04

2019  
2020  
2021  
2022 005464 000004  
2023 005466 012704 020722  
2024 005472 012705 005316  
2025 005476 004737 020524  
2026  
2027 005502 000400  
2028  
2029  
2030  
2031  
2032  
2033  
2034  
2035 005504 000004  
2036 005506 012704 020736  
2037 005512 012705 005020  
2038 005516 004737 020524  
2039  
2040 005522 000400  
2041  
2042  
2043  
2044  
2045  
2046  
2047 005524 000004  
2048 005526 012704 020736  
2049 005532 012705 005066  
2050 005536 004737 020524  
2051  
2052 005542 000400  
2053  
2054  
2055  
2056  
2057  
2058  
2059 005544 000004  
2060 005546 012704 020736  
2061 005552 012705 005134  
2062 005556 004737 020524  
2063  
2064 005562 000400  
2065  
2066  
2067  
2068  
2069  
2070  
2071 005564 000004  
2072 005566 012704 020736  
2073 005572 012705 005202  
2074 005576 004737 020524

```
*****  
*TEST 37 TEST FP AC1, D MODE, TEST PATTERN LDAD-6  
*****  
†ST37: SCOPE  
MOV #LDARD1,R4 ; PTR TO TESTING ROUTINE  
MOV #LDAD6,R5 ; PTR TO TEST DATA SET  
JSR PC,@#LDADT ; GO TEST  
  
BR TST40 ;  
  
; ( FOR DATA SEE TEST OF FP REG ACO WITH TEST PATTERN LDAD-6, ABOVE )
```

```
*****  
*TEST 40 TEST FP AC2, D MODE, TEST PATTERN LDAD-1  
*****  
†ST40: SCOPE  
MOV #LDARD2,R4 ; PTR TO TESTING ROUTINE  
MOV #LDAD1,R5 ; PTR TO TEST DATA SET  
JSR PC,@#LDADT ; GO TEST  
  
BR TST41 ;  
  
; ( FOR DATA SEE TEST OF FP REG ACO WITH TEST PATTERN LDAD-1, ABOVE )
```

```
*****  
*TEST 41 TEST FP AC2, D MODE, TEST PATTERN LDAD-2  
*****  
†ST41: SCOPE  
MOV #LDARD2,R4 ; PTR TO TESTING ROUTINE  
MOV #LDAD2,R5 ; PTR TO TEST DATA SET  
JSR PC,@#LDADT ; GO TEST  
  
BR TST42 ;  
  
; ( FOR DATA SEE TEST OF FP REG ACO WITH TEST PATTERN LDAD-2, ABOVE )
```

```
*****  
*TEST 42 TEST FP AC2, D MODE, TEST PATTERN LDAD-3  
*****  
†ST42: SCOPE  
MOV #LDARD2,R4 ; PTR TO TESTING ROUTINE  
MOV #LDAD3,R5 ; PTR TO TEST DATA SET  
JSR PC,@#LDADT ; GO TEST  
  
BR TST43 ;  
  
; ( FOR DATA SEE TEST OF FP REG ACO WITH TEST PATTERN LDAD-3, ABOVE )
```

```
*****  
*TEST 43 TEST FP AC2, D MODE, TEST PATTERN LDAD-4  
*****  
†ST43: SCOPE  
MOV #LDARD2,R4 ; PTR TO TESTING ROUTINE  
MOV #LDAD4,R5 ; PTR TO TEST DATA SET  
JSR PC,@#LDADT ; GO TEST
```

# F04

FPU BASIC INSTR TESTS MACY11 27(1006) 09-FEB-77 09:48 PAGE 46  
 DQFPA.P11 09-FEB-77 09:42 T43 TEST FP AC2, D MODE, TEST PATTERN LDAD-4

```

2075
2076 005602 000400          BR      TST44          ;;
2077
2078 ; ( FOR DATA SEE TEST OF FP REG ACO WITH TEST PATTERN LDAD-4, ABOVE )
2079
2080 ;:*****
2081 ;*TEST 44      TEST FP AC2, D MODE, TEST PATTERN LDAD-5
2082 ;:*****
2083 005604 000004          †ST44: SCOPE
2084 005606 012704 020736      MOV      #LDARD2,R4      ; PTR TO TESTING ROUTINE
2085 005612 012705 005250      MOV      #LDAD5,R5      ; PTR TO TEST DATA SET
2086 005616 004737 020524      JSR      PC,@#LDADT      ; GO TEST
2087
2088 005622 000400          BR      TST45          ;;
2089
2090 ; ( FOR DATA SEE TEST OF FP REG ACO WITH TEST PATTERN LDAD-5, ABOVE )
2091
2092 ;:*****
2093 ;*TEST 45      TEST FP AC2, D MODE, TEST PATTERN LDAD-6
2094 ;:*****
2095 005624 000004          †ST45: SCOPE
2096 005626 012704 020736      MOV      #LDARD2,R4      ; PTR TO TESTING ROUTINE
2097 005632 012705 005316      MOV      #LDAD6,R5      ; PTR TO TEST DATA SET
2098 005636 004737 020524      JSR      PC,@#LDADT      ; GO TEST
2099
2100 005642 000400          BR      TST46          ;;
2101
2102 ; ( FOR DATA SEE TEST OF FP REG ACO WITH TEST PATTERN LDAD-6, ABOVE )
2103
2104 ;:*****
2105 ;*TEST 46      TEST FP AC3, D MODE, TEST PATTERN LDAD-1
2106 ;:*****
2107 005644 000004          †ST46: SCOPE
2108 005646 012704 020752      MOV      #LDARD3,R4      ; PTR TO TESTING ROUTINE
2109 005652 012705 005020      MOV      #LDAD1,R5      ; PTR TO TEST DATA SET
2110 005656 004737 020524      JSR      PC,@#LDADT      ; GO TEST
2111
2112 005662 000400          BR      TST47          ;;
2113
2114 ; ( FOR DATA SEE TEST OF FP REG ACO WITH TEST PATTERN LDAD-1, ABOVE )
2115
2116 ;:*****
2117 ;*TEST 47      TEST FP AC3, D MODE, TEST PATTERN LDAD-2
2118 ;:*****
2119 005664 000004          †ST47: SCOPE
2120 005666 012704 020752      MOV      #LDARD3,R4      ; PTR TO TESTING ROUTINE
2121 005672 012705 005066      MOV      #LDAD2,R5      ; PTR TO TEST DATA SET
2122 005676 004737 020524      JSR      PC,@#LDADT      ; GO TEST
2123
2124 005702 000400          BR      TST50          ;;
2125
2126 ; ( FOR DATA SEE TEST OF FP REG ACO WITH TEST PATTERN LDAD-2, ABOVE )
2127
2128 ;:*****
2129 ;*TEST 50      TEST FP AC3, D MODE, TEST PATTERN LDAD-3
2130
  
```

2131  
2132 005704 000004  
2133 005706 012704 020752  
2134 005712 012705 005134  
2135 005716 004737 020524  
2136  
2137 005722 000400  
2138  
2139  
2140  
2141  
2142  
2143  
2144 005724 000004  
2145 005726 012704 020752  
2146 005732 012705 005202  
2147 005736 004737 020524  
2148  
2149 005742 000400  
2150  
2151  
2152  
2153  
2154  
2155  
2156 005744 000004  
2157 005746 012704 020752  
2158 005752 012705 005250  
2159 005756 004737 020524  
2160  
2161 005762 000400  
2162  
2163  
2164  
2165  
2166  
2167  
2168 005764 000004  
2169 005766 012704 020752  
2170 005772 012705 005316  
2171 005776 004737 020524  
2172  
2173 006002 000400  
2174  
2175  
2176  
2177  
2178  
2179  
2180  
2181 006004 000004  
2182 006006 012704 020766  
2183 006012 012705 005020  
2184 006016 004737 020524  
2185  
2186 006022 000400

```
*****  
TST50: SCOPE  
MOV #LDARD3,R4 ; PTR TO TESTING ROUTINE  
MOV #LDAD3,R5 ; PTR TO TEST DATA SET  
JSR PC,@#LDADT ; GO TEST  
BR TST51 ;;  
; ( FOR DATA SEE TEST OF FP REG ACO WITH TEST PATTERN LDAD-3, ABOVE )  
*****  
*TEST 51 TEST FP AC3, D MODE, TEST PATTERN LDAD-4  
*****  
TST51: SCOPE  
MOV #LDARD3,R4 ; PTR TO TESTING ROUTINE  
MOV #LDAD4,R5 ; PTR TO TEST DATA SET  
JSR PC,@#LDADT ; GO TEST  
BR TST52 ;;  
; ( FOR DATA SEE TEST OF FP REG ACO WITH TEST PATTERN LDAD-4, ABOVE )  
*****  
*TEST 52 TEST FP AC3, D MODE, TEST PATTERN LDAD-5  
*****  
TST52: SCOPE  
MOV #LDARD3,R4 ; PTR TO TESTING ROUTINE  
MOV #LDAD5,R5 ; PTR TO TEST DATA SET  
JSR PC,@#LDADT ; GO TEST  
BR TST53 ;;  
; ( FOR DATA SEE TEST OF FP REG ACO WITH TEST PATTERN LDAD-5, ABOVE )  
*****  
*TEST 53 TEST FP AC3, D MODE, TEST PATTERN LDAD-6  
*****  
TST53: SCOPE  
MOV #LDARD3,R4 ; PTR TO TESTING ROUTINE  
MOV #LDAD6,R5 ; PTR TO TEST DATA SET  
JSR PC,@#LDADT ; GO TEST  
BR TST54 ;;  
; ( FOR DATA SEE TEST OF FP REG ACO WITH TEST PATTERN LDAD-6, ABOVE )  
*****  
*TEST 54 TEST FP AC4, D MODE, TEST PATTERN LDAD-1  
*****  
TST54: SCOPE  
MOV #LDARD4,R4 ; PTR TO TESTING ROUTINE  
MOV #LDAD1,R5 ; PTR TO TEST DATA SET  
JSR PC,@#LDADT ; GO TEST  
BR TST55 ;;
```

H04

FPU BASIC INSTR TESTS MACY11 27(1006) 09-FEB-77 09:48 PAGE 48  
DQFPAA.P11 09-FEB-77 09:42

TS4 TEST FP AC4, D MODE, TEST PATTERN LDAD-1

2187  
2188  
2189  
2190  
2191  
2192  
2193 006024 000004  
2194 006026 012704 020766  
2195 006032 012705 005066  
2196 006036 004737 020524  
2197  
2198 006042 000400  
2199  
2200  
2201  
2202  
2203  
2204  
2205 006044 000004  
2206 006046 012704 020766  
2207 006052 012705 005134  
2208 006056 004737 020524  
2209  
2210 006062 000400  
2211  
2212  
2213  
2214  
2215  
2216  
2217 006064 000004  
2218 006066 012704 020766  
2219 006072 012705 005202  
2220 006076 004737 020524  
2221  
2222 006102 000400  
2223  
2224  
2225  
2226  
2227  
2228  
2229 006104 000004  
2230 006106 012704 020766  
2231 006112 012705 005250  
2232 006116 004737 020524  
2233  
2234 006122 000400  
2235  
2236  
2237  
2238  
2239  
2240  
2241  
2242 006124 000004

; ( FOR DATA SEE TEST OF FP REG ACO WITH TEST PATTERN LDAD-1, ABOVE )

\*\*\*\*\*  
; \*TEST 55 TEST FP AC4, D MODE, TEST PATTERN LDAD-2  
\*\*\*\*\*

TST55: SCOPE  
MOV #LDARD4,R4 ; PTR TO TESTING ROUTINE  
MOV #LDAD2,R5 ; PTR TO TEST DATA SET  
JSR PC,@#LDADT ; GO TEST  
BR TST56 ;;

; ( FOR DATA SEE TEST OF FP REG ACO WITH TEST PATTERN LDAD-2, ABOVE )

\*\*\*\*\*  
; \*TEST 56 TEST FP AC4, D MODE, TEST PATTERN LDAD-3  
\*\*\*\*\*

TST56: SCOPE  
MOV #LDARD4,R4 ; PTR TO TESTING ROUTINE  
MOV #LDAD3,R5 ; PTR TO TEST DATA SET  
JSR PC,@#LDADT ; GO TEST  
BR TST57 ;;

; ( FOR DATA SEE TEST OF FP REG ACO WITH TEST PATTERN LDAD-3, ABOVE )

\*\*\*\*\*  
; \*TEST 57 TEST FP AC4, D MODE, TEST PATTERN LDAD-4  
\*\*\*\*\*

TST57: SCOPE  
MOV #LDARD4,R4 ; PTR TO TESTING ROUTINE  
MOV #LDAD4,R5 ; PTR TO TEST DATA SET  
JSR PC,@#LDADT ; GO TEST  
BR TST60 ;;

; ( FOR DATA SEE TEST OF FP REG ACO WITH TEST PATTERN LDAD-4, ABOVE )

\*\*\*\*\*  
; \*TEST 60 TEST FP AC4, D MODE, TEST PATTERN LDAD-5  
\*\*\*\*\*

TST60: SCOPE  
MOV #LDARD4,R4 ; PTR TO TESTING ROUTINE  
MOV #LDAD5,R5 ; PTR TO TEST DATA SET  
JSR PC,@#LDADT ; GO TEST  
BR TST61 ;;

; ( FOR DATA SEE TEST OF FP REG ACO WITH TEST PATTERN LDAD-5, ABOVE )

\*\*\*\*\*  
; \*TEST 61 TEST FP AC5, D MODE, TEST PATTERN LDAD-1  
\*\*\*\*\*

TST61: SCOPE

```

FPU BASIC INSTR TESTS MACY11 27(1006) 09-FEB-77 09:48 PAGE 49
DQFPAA.P11 09-FEB-77 09:42 T61 TEST FP ACS, D MODE, TEST PATTERN LDAD-1

2243 006126 012704 021006      MOV      #LDARDS,R4      ; PTR TO TESTING ROUTINE
2244 006132 012705 005020      MOV      #LDAD1,R5      ; PTR TO TEST DATA SET
2245 006136 004737 020524      JSR      PC,@#LDADT     ; GO TEST
2246
2247 006142 000400              BR       TST62          ;;
2248
2249 ; ( FOR DATA SEE TEST OF FP REG ACO WITH TEST PATTERN LDAD-1, ABOVE )
2250
2251 ;:*****
2252 ;:TEST 62      TEST FP ACS, D MODE, TEST PATTERN LDAD-2
2253 ;:*****
2254 †TST62: SCOPE
2255 006144 000004              MOV      #LDARDS,R4      ; PTR TO TESTING ROUTINE
2256 006146 012704 021006      MOV      #LDAD2,R5      ; PTR TO TEST DATA SET
2257 006152 012705 005066      JSR      PC,@#LDADT     ; GO TEST
2258 006156 004737 020524
2259 006162 000400              BR       TST63          ;;
2260
2261 ; ( FOR DATA SEE TEST OF FP REG ACO WITH TEST PATTERN LDAD-2, ABOVE )
2262
2263 ;:*****
2264 ;:TEST 63      TEST FP ACS, D MODE, TEST PATTERN LDAD-3
2265 ;:*****
2266 †TST63: SCOPE
2267 006164 000004              MOV      #LDARDS,R4      ; PTR TO TESTING ROUTINE
2268 006166 012704 021006      MOV      #LDAD3,R5      ; PTR TO TEST DATA SET
2269 006172 012705 005134      JSR      PC,@#LDADT     ; GO TEST
2270 006176 004737 020524
2271 006202 000400              BR       TST64          ;;
2272
2273 ; ( FOR DATA SEE TEST OF FP REG ACO WITH TEST PATTERN LDAD-3, ABOVE )
2274
2275 ;:*****
2276 ;:TEST 64      TEST FP ACS, D MODE, TEST PATTERN LDAD-4
2277 ;:*****
2278 †TST64: SCOPE
2279 006204 000004              MOV      #LDARDS,R4      ; PTR TO TESTING ROUTINE
2280 006206 012704 021006      MOV      #LDAD4,R5      ; PTR TO TEST DATA SET
2281 006212 012705 005202      JSR      PC,@#LDADT     ; GO TEST
2282 006216 004737 020524
2283 006222 000400              BR       TST65          ;;
2284
2285 ; ( FOR DATA SEE TEST OF FP REG ACO WITH TEST PATTERN LDAD-4, ABOVE )
2286
2287 ;:*****
2288 ;:TEST 65      TEST FP ACS, D MODE, TEST PATTERN LDAD-5
2289 ;:*****
2290 †TST65: SCOPE
2291 006224 000004              MOV      #LDARDS,R4      ; PTR TO TESTING ROUTINE
2292 006226 012704 021006      MOV      #LDAD5,R5      ; PTR TO TEST DATA SET
2293 006232 012705 005250      JSR      PC,@#LDADT     ; GO TEST
2294 006236 004737 020524
2295 006242 000400              BR       TST66          ;;
2296
2297 ; ( FOR DATA SEE TEST OF FP REG ACO WITH TEST PATTERN LDAD-5, ABOVE )
2298

```



J04

FPU BASIC INSTR TESTS MACY11 27(1006) 09-FEB-77 09:48 PAGE 50  
DQFPAA.P11 09-FEB-77 09:42 T65 TEST FP ACS, D MODE, TEST PATTERN LDAD-5

2299  
2300

```

2301
2302
2303
2304 006244 000004
2305 006246 012704 021206
2306 006252 012705 006264
2307 006256 004737 021026
2308
2309 006262 000415
2310
2311 006264
2312 006264 000000 000000 000000
2313 006272 000000
2314 006274 177777 177777
2315 006300 177777 177777 000000
2316 006306 000000
2317 006310 047547 047750
2318 006314 000000
2319
2320
2321
2322
2323 006316 000004
2324 006320 012704 021206
2325 006324 012705 006336
2326 006330 004737 021026
2327
2328 006334 000415
2329
2330 006336
2331 006336 177777 177777 177777
2332 006344 177777
2333 006346 000000 000000
2334 006352 000000 000000 177777
2335 006360 177777
2336 006362 047413 047604
2337 006366 000000
2338
2339
2340
2341
2342 006370 000004
2343 006372 012704 021206
2344 006376 012705 006410
2345 006402 004737 021026
2346
2347 006406 000415
2348
2349 006410
2350 006410 052525 052525 052525
2351 006416 052525
2352 006420 125252 125252
2353 006424 125252 125252 052525
2354 006432 052525
2355 006434 047547 047750
2356 006440 000000

```

```

*****
;TEST 66 TEST FP ACO, F MODE, TEST PATTERN LDAF-1
*****
TST66: SCOPE
MOV #LDARFO,R4 ; PTR TO TESTING ROUTINE
MOV #LDAF1,R5 ; PTR TO TEST DATA SET
JSR PC,@#LDAFT ; GO TEST
BR TST67 ;;

LDAF1: ; TEST DATA SET LDAF-1:
.WORD 0,0,0,0 ; LDD PATTERN
.WORD M1,M1 ; LDF PATTERN
.WORD M1,M1,0,0 ; STD EXPECTED PATTERN
.WORD 047547,047750 ; FPS: BEFORE, AFTER
.WORD 000000 ; FEC AFTER ( 0 = N/A )

*****
;TEST 67 TEST FP ACO, F MODE, TEST PATTERN LDAF-2
*****
TST67: SCOPE
MOV #LDARFO,R4 ; PTR TO TESTING ROUTINE
MOV #LDAF2,R5 ; PTR TO TEST DATA SET
JSR PC,@#LDAFT ; GO TEST
BR TST70 ;;

LDAF2: ; TEST DATA SET LDAF-2:
.WORD M1,M1,M1,M1 ; LDD PATTERN
.WORD 0,0 ; LDF PATTERN
.WORD 0,0,M1,M1 ; STD EXPECTED PATTERN
.WORD 047413,047604 ; FPS: BEFORE, AFTER
.WORD 000000 ; FEC AFTER ( 0 = N/A )

*****
;TEST 70 TEST FP ACO, F MODE, TEST PATTERN LDAF-3
*****
TST70: SCOPE
MOV #LDARFO,R4 ; PTR TO TESTING ROUTINE
MOV #LDAF3,R5 ; PTR TO TEST DATA SET
JSR PC,@#LDAFT ; GO TEST
BR TST71 ;;

LDAF3: ; TEST DATA SET LDAF-3:
.WORD ALTP,ALTP,ALTP,ALTP ; LDD PATTERN
.WORD ALTN,ALTN ; LDF PATTERN
.WORD ALTN,ALTN,ALTP,ALTP ; STD EXPECTED PATTERN
.WORD 047547,047750 ; FPS: BEFORE, AFTER
.WORD 000000 ; FEC AFTER ( 0 = N/A )

```

```

2357
2358
2359
2360
2361 006442 000004
2362 006444 012704 021206
2363 006450 012705 006462
2364 006454 004737 021026
2365
2366 006460 000415
2367
2368 006462
2369 006462 125252 125252 125252
2370 006470 125252
2371 006472 052525 052525
2372 006476 052525 052525 125252
2373 006504 125252
2374 006506 047417 047600
2375 006512 000000
2376
2377
2378
2379
2380 006514 000004
2381 006516 012704 021206
2382 006522 012705 006534
2383 006526 004737 021026
2384
2385 006532 000415
2386
2387 006534
2388 006534 177777 177777 177777
2389 006542 177777
2390 006544 100000 000000
2391 006550 100000 000000 177777
2392 006556 177777
2393 006560 043443 043654
2394 006564 000000
2395
2396
2397
2398
2399 006566 000004
2400 006570 012704 021206
2401 006574 012705 006606
2402 006600 004737 021026
2403
2404 006604 000415
2405
2406 006606
2407 006606 052525 000000 125252
2408 006614 177777
2409 006616 100000 000000
2410 006622 052525 000000 125252
2411 006630 177777
2412 006632 047543 147754

*****
*TEST 71 TEST FP ACO, F MODE, TEST PATTERN LDAF-4
*****
TST71: SCOPE
MOV #LDAF0,R4 ; PTR TO TESTING ROUTINE
MOV #LDAF4,R5 ; PTR TO TEST DATA SET
JSR PC,@#LDAFT ; GO TEST
BR TST72 ;;

LDAF4: ; TEST DATA SET LDAF-4:
.WORD ALTN,ALTN,ALTN,ALTN ; LDD PATTERN
.WORD ALTP,ALTP ; LDF PATTERN
.WORD ALTP,ALTP,ALTN,ALTN ; STD EXPECTED PATTERN
.WORD 047417,047600 ; FPS: BEFORE, AFTER
.WORD 000000 ; FEC AFTER ( 0 = N/A )

*****
*TEST 72 TEST FP ACO, F MODE, TEST PATTERN LDAF-5
*****
TST72: SCOPE
MOV #LDAF0,R4 ; PTR TO TESTING ROUTINE
MOV #LDAF5,R5 ; PTR TO TEST DATA SET
JSR PC,@#LDAFT ; GO TEST
BR TST73 ;;

LDAF5: ; TEST DATA SET LDAF-5:
.WORD M1,M1,M1,M1 ; LDD PATTERN
.WORD M0,00 ; LDF PATTERN
.WORD M0,00,M1,M1 ; STD EXPECTED PATTERN
.WORD 043443,043654 ; FPS: BEFORE, AFTER
.WORD 000000 ; FEC AFTER ( 0 = N/A )

*****
*TEST 73 TEST FP ACO, F MODE, TEST PATTERN LDAF-6
*****
TST73: SCOPE
MOV #LDAF0,R4 ; PTR TO TESTING ROUTINE
MOV #LDAF6,R5 ; PTR TO TEST DATA SET
JSR PC,@#LDAFT ; GO TEST
BR TST74 ;;

LDAF6: ; TEST DATA SET LDAF-6:
.WORD ALTP,0,ALTN,M1 ; LDD PATTERN
.WORD M0,0 ; LDF PATTERN
.WORD ALTP,0,ALTN,M1 ; STD EXPECTED PATTERN
.WORD 047543,147754 ; FPS: BEFORE, AFTER

```

M04

13 006636 100014  
14  
15  
16  
17  
18  
19 006640 000004  
20 006642 012704 021230  
21 006646 012705 006264  
22 006652 004737 021026  
23  
24 006656 000400  
25  
26  
27  
28  
29  
30  
31 006660 000004  
32 006662 012704 021230  
33 006666 012705 006336  
34 006672 004737 021026  
35  
36 006676 000400  
37  
38  
39  
40  
41  
42  
43 006700 000004  
44 006702 012704 021230  
45 006706 012705 006410  
46 006712 004737 021026  
47  
48 006716 000400  
49  
50  
51  
52  
53  
54  
55 006720 000004  
56 006722 012704 021230  
57 006726 012705 006462  
58 006732 004737 021026  
59  
60 006736 000400  
61  
62  
63  
64  
65  
66  
67 006740 000004  
68 006742 012704 021230

```
.WORD 100014 ; FEC AFTER ( 0 = N/A )

; *****
; *TEST 74 TEST FP AC1, F MODE, TEST PATTERN LDAF-1
; *****
TST74: SCOPE
        MOV #LDAF1,R4 ; PTR TO TESTING ROUTINE
        MOV #LDAF1,R5 ; PTR TO TEST DATA SET
        JSR PC,@#LDAFT ; GO TEST
        BR TST75 ;;
; ( FOR DATA SEE TEST OF FP REG ACO WITH TEST PATTERN LDAF-1, ABOVE )

; *****
; *TEST 75 TEST FP AC1, F MODE, TEST PATTERN LDAF-2
; *****
TST75: SCOPE
        MOV #LDAF1,R4 ; PTR TO TESTING ROUTINE
        MOV #LDAF2,R5 ; PTR TO TEST DATA SET
        JSR PC,@#LDAFT ; GO TEST
        BR TST76 ;;
; ( FOR DATA SEE TEST OF FP REG ACO WITH TEST PATTERN LDAF-2, ABOVE )

; *****
; *TEST 76 TEST FP AC1, F MODE, TEST PATTERN LDAF-3
; *****
TST76: SCOPE
        MOV #LDAF1,R4 ; PTR TO TESTING ROUTINE
        MOV #LDAF3,R5 ; PTR TO TEST DATA SET
        JSR PC,@#LDAFT ; GO TEST
        BR TST77 ;;
; ( FOR DATA SEE TEST OF FP REG ACO WITH TEST PATTERN LDAF-3, ABOVE )

; *****
; *TEST 77 TEST FP AC1, F MODE, TEST PATTERN LDAF-4
; *****
TST77: SCOPE
        MOV #LDAF1,R4 ; PTR TO TESTING ROUTINE
        MOV #LDAF4,R5 ; PTR TO TEST DATA SET
        JSR PC,@#LDAFT ; GO TEST
        BR TST100 ;;
; ( FOR DATA SEE TEST OF FP REG ACO WITH TEST PATTERN LDAF-4, ABOVE )

; *****
; *TEST 100 TEST FP AC1, F MODE, TEST PATTERN LDAF-5
; *****
TST100: SCOPE
        MOV #LDAF1,R4 ; PTR TO TESTING ROUTINE
```

N04

FPU BASIC INSTR TESTS MACY11 27(1006) 09-FEB-77 09:48 PAGE 54  
DQFPRA.P11 09-FEB-77 09:42

T100 TEST FP AC1, F MODE, TEST PATTERN LDAF-5

2469 006746 012705 006534  
2470 006752 004737 021026

MOV #LDAF5,R5 ; PTR TO TEST DATA SET  
JSR PC,@#LDAFT ; GO TEST

2472 006756 000400

BR TST101 ;;

; ( FOR DATA SEE TEST OF FP REG ACO WITH TEST PATTERN LDAF-5, ABOVE )

\*\*\*\*\*  
; \*TEST 101 TEST FP AC1, F MODE, TEST PATTERN LDAF-6  
\*\*\*\*\*

2479 006760 000004  
2480 006762 012704 021230  
2481 006766 012705 006606  
2482 006772 004737 021026

TST101: SCOPE  
MOV #LDARF1,R4 ; PTR TO TESTING ROUTINE  
MOV #LDAF6,R5 ; PTR TO TEST DATA SET  
JSR PC,@#LDAFT ; GO TEST

2484 006776 000400

BR TST102 ;;

; ( FOR DATA SEE TEST OF FP REG ACO WITH TEST PATTERN LDAF-6, ABOVE )

\*\*\*\*\*  
; \*TEST 102 TEST FP AC2, F MODE, TEST PATTERN LDAF-1  
\*\*\*\*\*

2492 007000 000004  
2493 007002 012704 021252  
2494 007006 012705 006264  
2495 007012 004737 021026

TST102: SCOPE  
MOV #LDARF2,R4 ; PTR TO TESTING ROUTINE  
MOV #LDAF1,R5 ; PTR TO TEST DATA SET  
JSR PC,@#LDAFT ; GO TEST

2497 007016 000400

BR TST103 ;;

; ( FOR DATA SEE TEST OF FP REG ACO WITH TEST PATTERN LDAF-1, ABOVE )

\*\*\*\*\*  
; \*TEST 103 TEST FP AC2, F MODE, TEST PATTERN LDAF-2  
\*\*\*\*\*

2504 007020 000004  
2505 007022 012704 021252  
2506 007026 012705 006336  
2507 007032 004737 021026

TST103: SCOPE  
MOV #LDARF2,R4 ; PTR TO TESTING ROUTINE  
MOV #LDAF2,R5 ; PTR TO TEST DATA SET  
JSR PC,@#LDAFT ; GO TEST

2509 007036 000400

BR TST104 ;;

; ( FOR DATA SEE TEST OF FP REG ACO WITH TEST PATTERN LDAF-2, ABOVE )

\*\*\*\*\*  
; \*TEST 104 TEST FP AC2, F MODE, TEST PATTERN LDAF-3  
\*\*\*\*\*

2516 007040 000004  
2517 007042 012704 021252  
2518 007046 012705 006410  
2519 007052 004737 021026

TST104: SCOPE  
MOV #LDARF2,R4 ; PTR TO TESTING ROUTINE  
MOV #LDAF3,R5 ; PTR TO TEST DATA SET  
JSR PC,@#LDAFT ; GO TEST

2521 007056 000400

BR TST105 ;;

; ( FOR DATA SEE TEST OF FP REG ACO WITH TEST PATTERN LDAF-3, ABOVE )

2522  
2523  
2524

2525  
2526  
2527  
2528  
2529  
2530  
2531  
2532  
2533  
2534  
2535  
2536  
2537  
2538  
2539  
2540  
2541  
2542  
2543  
2544  
2545  
2546  
2547  
2548  
2549  
2550  
2551  
2552  
2553  
2554  
2555  
2556  
2557  
2558  
2559  
2560  
2561  
2562  
2563  
2564  
2565  
2566  
2567  
2568  
2569  
2570  
2571  
2572  
2573  
2574  
2575  
2576  
2577  
2578  
2579  
2580

007060 000004  
007062 012704 021252  
007066 012705 006462  
007072 004737 021026  
  
007076 000400  
  
  
  
  
  
  
  
  
  
007100 000004  
007102 012704 021252  
007106 012705 006534  
007112 004737 021026  
  
007116 000400  
  
  
  
  
  
  
  
  
  
007120 000004  
007122 012704 021252  
007126 012705 006606  
007132 004737 021026  
  
007136 000400  
  
  
  
  
  
  
  
  
  
007140 000004  
007142 012704 021274  
007146 012705 006264  
007152 004737 021026  
  
007156 000400  
  
  
  
  
  
  
  
  
  
007160 000004  
007162 012704 021274  
007166 012705 006336  
007172 004737 021026

```
*****  
*TEST 105 TEST FP AC2, F MODE, TEST PATTERN LDAF-4  
*****  
TST105: SCOPE  
MOV #LDAF2,R4 ; PTR TO TESTING ROUTINE  
MOV #LDAF4,R5 ; PTR TO TEST DATA SET  
JSR PC,@#LDAFT ; GO TEST  
  
BR TST106 ;;  
  
; ( FOR DATA SEE TEST OF FP REG AC0 WITH TEST PATTERN LDAF-4, ABOVE )  
  
*****  
*TEST 106 TEST FP AC2, F MODE, TEST PATTERN LDAF-5  
*****  
TST106: SCOPE  
MOV #LDAF2,R4 ; PTR TO TESTING ROUTINE  
MOV #LDAF5,R5 ; PTR TO TEST DATA SET  
JSR PC,@#LDAFT ; GO TEST  
  
BR TST107 ;;  
  
; ( FOR DATA SEE TEST OF FP REG AC0 WITH TEST PATTERN LDAF-5, ABOVE )  
  
*****  
*TEST 107 TEST FP AC2, F MODE, TEST PATTERN LDAF-6  
*****  
TST107: SCOPE  
MOV #LDAF2,R4 ; PTR TO TESTING ROUTINE  
MOV #LDAF6,R5 ; PTR TO TEST DATA SET  
JSR PC,@#LDAFT ; GO TEST  
  
BR TST110 ;;  
  
; ( FOR DATA SEE TEST OF FP REG AC0 WITH TEST PATTERN LDAF-6, ABOVE )  
  
*****  
*TEST 110 TEST FP AC3, F MODE, TEST PATTERN LDAF-1  
*****  
TST110: SCOPE  
MOV #LDAF3,R4 ; PTR TO TESTING ROUTINE  
MOV #LDAF1,R5 ; PTR TO TEST DATA SET  
JSR PC,@#LDAFT ; GO TEST  
  
BR TST111 ;;  
  
; ( FOR DATA SEE TEST OF FP REG AC0 WITH TEST PATTERN LDAF-1, ABOVE )  
  
*****  
*TEST 111 TEST FP AC3, F MODE, TEST PATTERN LDAF-2  
*****  
TST111: SCOPE  
MOV #LDAF3,R4 ; PTR TO TESTING ROUTINE  
MOV #LDAF2,R5 ; PTR TO TEST DATA SET  
JSR PC,@#LDAFT ; GO TEST
```

```

2581
2582 007176 000400          BR      TST112          ;;
2583
2584          ; ( FOR DATA SEE TEST OF FP REG AC0 WITH TEST PATTERN LDAF-2, ABOVE )
2585
2586          ;:*****
2587          ;:TEST 112      TEST FP AC3, F MODE, TEST PATTERN LDAF-3
2588          ;:*****
2589 007200 000004          †TST112: SCOPE
2590 007202 012704 021274      MOV      #LDAF3,R4          ; PTR TO TESTING ROUTINE
2591 007206 012705 006410      MOV      #LDAF3,R5          ; PTR TO TEST DATA SET
2592 007212 004737 021026      JSR      PC,@#LDAFT        ; GO TEST
2593
2594 007216 000400          BR      TST113          ;;
2595
2596          ; ( FOR DATA SEE TEST OF FP REG AC0 WITH TEST PATTERN LDAF-3, ABOVE )
2597
2598          ;:*****
2599          ;:TEST 113      TEST FP AC3, F MODE, TEST PATTERN LDAF-4
2600          ;:*****
2601 007220 000004          †TST113: SCOPE
2602 007222 012704 021274      MOV      #LDAF3,R4          ; PTR TO TESTING ROUTINE
2603 007226 012705 006462      MOV      #LDAF4,R5          ; PTR TO TEST DATA SET
2604 007232 004737 021026      JSR      PC,@#LDAFT        ; GO TEST
2605
2606 007236 000400          BR      TST114          ;;
2607
2608          ; ( FOR DATA SEE TEST OF FP REG AC0 WITH TEST PATTERN LDAF-4, ABOVE )
2609
2610          ;:*****
2611          ;:TEST 114      TEST FP AC3, F MODE, TEST PATTERN LDAF-5
2612          ;:*****
2613 007240 000004          †TST114: SCOPE
2614 007242 012704 021274      MOV      #LDAF3,R4          ; PTR TO TESTING ROUTINE
2615 007246 012705 006534      MOV      #LDAF5,R5          ; PTR TO TEST DATA SET
2616 007252 004737 021026      JSR      PC,@#LDAFT        ; GO TEST
2617
2618 007256 000400          BR      TST115          ;;
2619
2620          ; ( FOR DATA SEE TEST OF FP REG AC0 WITH TEST PATTERN LDAF-5, ABOVE )
2621
2622          ;:*****
2623          ;:TEST 115      TEST FP AC3, F MODE, TEST PATTERN LDAF-6
2624          ;:*****
2625 007260 000004          †TST115: SCOPE
2626 007262 012704 021274      MOV      #LDAF3,R4          ; PTR TO TESTING ROUTINE
2627 007266 012705 006606      MOV      #LDAF6,R5          ; PTR TO TEST DATA SET
2628 007272 004737 021026      JSR      PC,@#LDAFT        ; GO TEST
2629
2630 007276 000400          BR      TST116          ;;
2631
2632          ; ( FOR DATA SEE TEST OF FP REG AC0 WITH TEST PATTERN LDAF-6, ABOVE )
2633
2634          ;:*****
2635          ;:TEST 116      TEST FP AC4, F MODE, TEST PATTERN LDAF-1
2636

```

2637  
 2638 007300 000004  
 2639 007302 012704 021316  
 2640 007306 012705 006264  
 2641 007312 004737 021026  
 2642  
 2643 007316 000400  
 2644  
 2645  
 2646  
 2647  
 2648  
 2649  
 2650 007320 000004  
 2651 007322 012704 021316  
 2652 007326 012705 006336  
 2653 007332 004737 021026  
 2654  
 2655 007336 000400  
 2656  
 2657  
 2658  
 2659  
 2660  
 2661  
 2662 007340 000004  
 2663 007342 012704 021316  
 2664 007346 012705 006410  
 2665 007352 004737 021026  
 2666  
 2667 007356 000400  
 2668  
 2669  
 2670  
 2671  
 2672  
 2673  
 2674 007360 000004  
 2675 007362 012704 021316  
 2676 007366 012705 006462  
 2677 007372 004737 021026  
 2678  
 2679 007376 000400  
 2680  
 2681  
 2682  
 2683  
 2684  
 2685  
 2686 007400 000004  
 2687 007402 012704 021316  
 2688 007406 012705 006534  
 2689 007412 004737 021026  
 2690  
 2691 007416 000400  
 2692

```

:*****
†TST116: SCOPE
      MOV      #LDARF4,R4      ; PTR TO TESTING ROUTINE
      MOV      #LDAF1,R5      ; PTR TO TEST DATA SET
      JSR      PC,@#LDAFT     ; GO TEST
      BR       TST117        ;;
; ( FOR DATA SEE TEST OF FP REG ACO WITH TEST PATTERN LDAF-1, ABOVE )
:*****
; *TEST 117      TEST FP AC4, F MODE, TEST PATTERN LDAF-2
:*****
†TST117: SCOPE
      MOV      #LDARF4,R4      ; PTR TO TESTING ROUTINE
      MOV      #LDAF2,R5      ; PTR TO TEST DATA SET
      JSR      PC,@#LDAFT     ; GO TEST
      BR       TST120        ;;
; ( FOR DATA SEE TEST OF FP REG ACO WITH TEST PATTERN LDAF-2, ABOVE )
:*****
; *TEST 120      TEST FP AC4, F MODE, TEST PATTERN LDAF-3
:*****
†TST120: SCOPE
      MOV      #LDARF4,R4      ; PTR TO TESTING ROUTINE
      MOV      #LDAF3,R5      ; PTR TO TEST DATA SET
      JSR      PC,@#LDAFT     ; GO TEST
      BR       TST121        ;;
; ( FOR DATA SEE TEST OF FP REG ACO WITH TEST PATTERN LDAF-3, ABOVE )
:*****
; *TEST 121      TEST FP AC4, F MODE, TEST PATTERN LDAF-4
:*****
†TST121: SCOPE
      MOV      #LDARF4,R4      ; PTR TO TESTING ROUTINE
      MOV      #LDAF4,R5      ; PTR TO TEST DATA SET
      JSR      PC,@#LDAFT     ; GO TEST
      BR       TST122        ;;
; ( FOR DATA SEE TEST OF FP REG ACO WITH TEST PATTERN LDAF-4, ABOVE )
:*****
; *TEST 122      TEST FP AC4, F MODE, TEST PATTERN LDAF-5
:*****
†TST122: SCOPE
      MOV      #LDARF4,R4      ; PTR TO TESTING ROUTINE
      MOV      #LDAF5,R5      ; PTR TO TEST DATA SET
      JSR      PC,@#LDAFT     ; GO TEST
      BR       TST123        ;;

```



E05

FPU BASIC INSTR TESTS MACY11 27(1006) 09-FEB-77 09:48 PAGE 58  
DQFPAA.P11 09-FEB-77 09:42

T122 TEST FP AC4, F MODE, TEST PATTERN LDAF-5

; ( FOR DATA SEE TEST OF FP REG ACO WITH TEST PATTERN LDAF-5, ABOVE )

2693  
2694  
2695  
2696  
2697  
2698  
2699 007420 000004  
2700 007422 012704 021346  
2701 007426 012705 006264  
2702 007432 004737 021026  
2703  
2704 007436 000400

\*\*\*\*\*  
; \*TEST 123 TEST FP ACS, F MODE, TEST PATTERN LDAF-1  
\*\*\*\*\*  
†TST123: SCOPE  
MOV #LDARF5,R4 ; PTR TO TESTING ROUTINE  
MOV #LDAF1,R5 ; PTR TO TEST DATA SET  
JSR PC,@#LDAFT ; GO TEST  
BR TST124 ;;

; ( FOR DATA SEE TEST OF FP REG ACO WITH TEST PATTERN LDAF-1, ABOVE )

2705  
2706  
2707  
2708  
2709  
2710  
2711 007440 000004  
2712 007442 012704 021346  
2713 007446 012705 006336  
2714 007452 004737 021026  
2715  
2716 007456 000400

\*\*\*\*\*  
; \*TEST 124 TEST FP ACS, F MODE, TEST PATTERN LDAF-2  
\*\*\*\*\*  
†TST124: SCOPE  
MOV #LDARF5,R4 ; PTR TO TESTING ROUTINE  
MOV #LDAF2,R5 ; PTR TO TEST DATA SET  
JSR PC,@#LDAFT ; GO TEST  
BR TST125 ;;

; ( FOR DATA SEE TEST OF FP REG ACO WITH TEST PATTERN LDAF-2, ABOVE )

2717  
2718  
2719  
2720  
2721  
2722  
2723 007460 000004  
2724 007462 012704 021346  
2725 007466 012705 006410  
2726 007472 004737 021026  
2727  
2728 007476 000400

\*\*\*\*\*  
; \*TEST 125 TEST FP ACS, F MODE, TEST PATTERN LDAF-3  
\*\*\*\*\*  
†TST125: SCOPE  
MOV #LDARF5,R4 ; PTR TO TESTING ROUTINE  
MOV #LDAF3,R5 ; PTR TO TEST DATA SET  
JSR PC,@#LDAFT ; GO TEST  
BR TST126 ;;

; ( FOR DATA SEE TEST OF FP REG ACO WITH TEST PATTERN LDAF-3, ABOVE )

2729  
2730  
2731  
2732  
2733  
2734  
2735 007500 000004  
2736 007502 012704 021346  
2737 007506 012705 006462  
2738 007512 004737 021026  
2739  
2740 007516 000400

\*\*\*\*\*  
; \*TEST 126 TEST FP ACS, F MODE, TEST PATTERN LDAF-4  
\*\*\*\*\*  
†TST126: SCOPE  
MOV #LDARF5,R4 ; PTR TO TESTING ROUTINE  
MOV #LDAF4,R5 ; PTR TO TEST DATA SET  
JSR PC,@#LDAFT ; GO TEST  
BR TST127 ;;

; ( FOR DATA SEE TEST OF FP REG ACO WITH TEST PATTERN LDAF-4, ABOVE )

2741  
2742  
2743  
2744  
2745  
2746  
2747 007520 000004  
2748 007522 012704 021346

\*\*\*\*\*  
; \*TEST 127 TEST FP ACS, F MODE, TEST PATTERN LDAF-5  
\*\*\*\*\*  
†TST127: SCOPE  
MOV #LDARF5,R4 ; PTR TO TESTING ROUTINE

# F05

FPU BASIC INSTR TESTS MACY11 27(1006) 09-FEB-77 09:48 PAGE 59  
DQFPAA.P11 09-FEB-77 09:42 T127 TEST FP ACS, F MODE, TEST PATTERN LDAF-5

2749	007526	012705	006534	MOV	#LDAF5,R5	; PTR TO TEST DATA SET
2750	007532	004737	021026	JSR	PC,@#LDAFT	; GO TEST
2751						
2752	007536	000400		BR	TST130	::
2753						

2754 ; ( FOR DATA SEE TEST OF FP REG ACO WITH TEST PATTERN LDAF-5, ABOVE )  
2755  
2756  
2757

```

2758
2759
2760
2761
2762
2763
2764
2765
2766 007540 000004
2767 007542 170127 047600
2768 007546 016767 000102 171414
2769 007554 016767 000076 171410
2770 007562 016767 000072 171404
2771 007570 016767 000066 171400
2772
2773 007576 012705 001170
2774 007602 172415
2775
2776 007604 174067 171370
2777
2778 007610 026767 171354 171362
2779 007616 001014
2780 007620 026767 171346 171354
2781 007626 001010
2782 007630 026767 171340 171346
2783 007636 001004
2784 007640 026767 171332 171340
2785 007646 001401
2786 007650 104031
2787
2788 007652
2789 007652 000404
2790
2791 007654 052525 052525 052525
2792 007662 052525
2793
2794
2795
2796
2797
2798 007664 000004
2799 007666 170127 047600
2800 007672 016767 000112 171270
2801 007700 016767 000106 171264
2802 007706 016767 000102 171260
2803 007714 016767 000076 171254
2804
2805 007722 012704 001200
2806 007726 172544
2807
2808 007730 174137 001200
2809
2810 007734 020427 001170
2811 007740 001401
2812 007742 104032
2813 007744

```

```

;*****
;.DSABL AMA ; ASSEMBLE ALL REFERENCES AS THEY ARE PRINTED
;*****

```

```

;*****
;TEST 130 TEST OF LOAD-FSRC MODE-1, STORE-FDST MODE-6(PC), D MODE
;*****

```

```

TST130: SCOPE
LDFPS #047600 ; D MODE FPS
MOV ADRD1+0,$REG0 ; MOVE PATTERN
MOV ADRD1+2,$REG1
MOV ADRD1+4,$REG2
MOV ADRD1+6,$REG3
MOV #SREG0,R5 ; ADDR(DATA)
LDD (R5),AC0 ; M1-R5
STD AC0,$REG4 ; M6-R7
CMP $REG0,$REG4 ; 1ST WORD PATTERN CHECK?
BNE 64$
CMP $REG1,$REG5 ; 2ND WORD PATTERN CHECK?
BNE 64$
CMP $REG2,$REG6 ; 3RD WORD PATTERN CHECK?
BNE 64$
CMP $REG3,$REG7 ; 4TH WORD PATTERN CHECK?
BEQ 65$
64$: ERROR 31 ; PATTERN DOESNT MATCH

```

```

65$: BR TST131 ;;
ADR1: .WORD ALTP,ALTP,ALTP,ALTP ; TEST PATTERN

```

```

;*****
;TEST 131 TEST OF LOAD-FSRC MODE-4, STORE-FDST MODE-3(PC), D MODE
;*****

```

```

TST131: SCOPE
LDFPS #047600 ; D MODE FPS
MOV ADRD2+0,$REG0 ; MOVE PATTERN
MOV ADRD2+2,$REG1
MOV ADRD2+4,$REG2
MOV ADRD2+6,$REG3
MOV #SREG0+10,R4 ; ADDR(DATA+10)
LDD -(R4),AC1 ; M4-R4
STD AC1,@#SREG4 ; M3-R7
CMP R4,$SREG0 ; WAS FSRC ADDR REG DECRE RIGHT AMOUNT?
BEQ 66$
66$: ERROR 32 ; NOT EQUAL, SIGNAL ERROR

```

66\$:

```

2814
2815 007744 026767 171220 171226      CMP      $REG0,$REG4      ; 1ST WORD PATTERN CHECK?
2816 007752 001014                    BNE      64$              ;
2817 007754 026767 171212 171220      CMP      $REG1,$REG5      ; 2ND WORD PATTERN CHECK?
2818 007762 001010                    BNE      64$              ;
2819 007764 026767 171204 171212      CMP      $REG2,$REG6      ; 3RD WORD PATTERN CHECK?
2820 007772 001004                    BNE      64$              ;
2821 007774 026767 171176 171204      CMP      $REG3,$REG7      ; 4TH WORD PATTERN CHECK?
2822 010002 001401                    BEQ      65$              ;
2823 010004 104033                    64$:  ERROR      33      ; PATTERN DOESNT MATCH
2824
2825 010006                    65$:
2826 010006 000404                    BR      TST132          ;;
2827
2828 010010 170360 170360 170360  ADDR2: .WORD  ALT4N,ALT4N,ALT4N,ALT4N ; TEST PATTERN
2829 010016 170360
2830
2831
2832
2833
2834

```

```

*****
*TEST 132 TEST OF LOAD-FSRC MODE-7, STORE-FDST MODE-3, D MODE
*****

```

```

2835 010020 000004                    TST132: SCOPE
2836 010022 170127 047600              LDFPS   #047600          ; D MODE FPS
2837 010026 016767 000132 171134      MOV     ADDR3+0,$REG0     ; MOVE PATTERN
2838 010034 016767 000126 171130      MOV     ADDR3+2,$REG1     ;
2839 010042 016767 000122 171124      MOV     ADDR3+4,$REG2     ;
2840 010050 016767 000116 171120      MOV     ADDR3+6,$REG3     ;
2841
2842 010056 012767 001170 171124      MOV     #$REG0,$REG10     ; ADDR(DATA)
2843 010064 012705 001214              MOV     #$REG10+4,R5     ; ADDR(ADDR(DATA))+4
2844 010070 172675 177774              LDD     2-4(R5),AC2       ; M7-R5
2845
2846 010074 012767 001200 171110      MOV     #$REG4,$REG11     ; ADDR(DEST)
2847 010102 012704 001212              MOV     #$REG11,R4        ; ADDR(ADDR(DEST))
2848 010106 174234                    STD     AC2,2(R4)+        ; M3-R4
2849

```

```

2850 010110 020427 001214              CMP     R4,$REG11+2       ; WAS FDST ADDR REG INCRE RIGHT AMOUNT?
2851 010114 001401                    BEQ     66$               ;
2852 010116 104034                    66$:  ERROR      34      ; NOT EQUAL, SIGNAL ERROR
2853 010120
2854

```

```

2855 010120 026767 171044 171052      CMP     $REG0,$REG4      ; 1ST WORD PATTERN CHECK?
2856 010126 001014                    BNE     64$              ;
2857 010130 026767 171036 171044      CMP     $REG1,$REG5      ; 2ND WORD PATTERN CHECK?
2858 010136 001010                    BNE     64$              ;
2859 010140 026767 171030 171036      CMP     $REG2,$REG6      ; 3RD WORD PATTERN CHECK?
2860 010146 001004                    BNE     64$              ;
2861 010150 026767 171022 171030      CMP     $REG3,$REG7      ; 4TH WORD PATTERN CHECK?
2862 010156 001401                    BEQ     65$              ;
2863 010160 104035                    64$:  ERROR      35      ; PATTERN DOESNT MATCH
2864

```

```

2865 010162                    65$:
2866 010162 000404                    BR      TST133          ;;
2867
2868 010164 125252 125252 125252  ADDR3: .WORD  ALTN,ALTN,ALTN,ALTN ; TEST PATTERN
2869 010172 125252

```

```

2870
2871
2872
2873
2874
2875 010174 000004
2876 010176 170127 047600
2877 010202 016767 000112 170760
2878 010210 016767 000106 170754
2879 010216 016767 000102 170750
2880 010224 016767 000076 170744
2881
2882 010232 172767 170732
2883
2884 010236 012705 001200
2885 010242 174325
2886
2887 010244 020527 001210
2888 010250 001401
2889 010252 104036
2890 010254
2891
2892 010254 026767 170710 170716
2893 010262 001014
2894 010264 026767 170702 170710
2895 010272 001010
2896 010274 026767 170674 170702
2897 010302 001004
2898 010304 026767 170666 170674
2899 010312 001401
2900 010314 104031
2901
2902 010316
2903 010316 000404
2904
2905 010320 007417 007417 007417 ADDR4: .WORD ALT4P,ALT4P,ALT4P,ALT4P ; TEST PATTERN
2906 010326 007417
2907
2908
2909
2910
2911
2912 010330 000004
2913 010332 170127 047600
2914 010336 016767 000114 170624
2915 010344 016767 000110 170620
2916 010352 016767 000104 170614
2917 010360 016767 000100 170610
2918
2919 010366 012700 001170
2920 010372 172420
2921
2922 010374 012703 001200
2923 010400 174013
2924
2925 010402 020027 001200

```

```

*****
;*TEST 133 TEST OF LOAD-FSRC MODE-6(PC), STORE-FDST MODE-2, D MODE
*****
↑T133: SCOPE
LDFPS #047600 ; D MODE FPS
MOV ADDR4+0,$REG0 ; MOVE PATTERN
MOV ADDR4+2,$REG1
MOV ADDR4+4,$REG2
MOV ADDR4+6,$REG3
LDD $REG0,AC3 ; M6-R7
MOV #REG4,R5 ; ADDR(DEST)
STD AC3,(R5)+ ; M2-R5
CMP R5,#REG4+10 ; WAS FDST ADDR REG INCRE RIGHT AMOUNT?
BEQ 66$
ERROR 36 ; NOT EQUAL, SIGNAL ERROR
66$:
CMP $REG0,$REG4 ; 1ST WORD PATTERN CHECK?
BNE 64$
CMP $REG1,$REG5 ; 2ND WORD PATTERN CHECK?
BNE 64$
CMP $REG2,$REG6 ; 3RD WORD PATTERN CHECK?
BNE 64$
CMP $REG3,$REG7 ; 4TH WORD PATTERN CHECK?
BEQ 65$
ERROR 31 ; PATTERN DOESNT MATCH
64$:
65$:
BR TST134 ;;

```

```

*****
;*TEST 134 TEST OF LOAD-FSRC MODE-2, STORE-FDST MODE-1, D MODE
*****
↑T134: SCOPE
LDFPS #047600 ; D MODE FPS
MOV ADDR5+0,$REG0 ; MOVE PATTERN
MOV ADDR5+2,$REG1
MOV ADDR5+4,$REG2
MOV ADDR5+6,$REG3
MOV #REG0,R0 ; ADDR(DATA)
LDD (R0)+,AC0 ; M2-R0
MOV #REG4,R3 ; ADDR(DEST)
STD AC0,(R3) ; M1-R3
CMP R0,#REG0+10 ; WAS FSRC ADDR REG INCRE RIGHT AMOUNT?

```

```

2926 010406 001401      BEQ      66$      ;
2927 010410 104037      ERROR    37      ; NOT EQUAL, SIGNAL ERROR
2928 010412      66$:
2929
2930 010412 026767 170552 170560      CMP      $REG0,$REG4 ; 1ST WORD PATTERN CHECK?
2931 010420 001014      BNE     64$      ;
2932 010422 026767 170544 170552      CMP      $REG1,$REG5 ; 2ND WORD PATTERN CHECK?
2933 010430 001010      BNE     64$      ;
2934 010432 026767 170536 170544      CMP      $REG2,$REG6 ; 3RD WORD PATTERN CHECK?
2935 010440 001004      BNE     64$      ;
2936 010442 026767 170530 170536      CMP      $REG3,$REG7 ; 4TH WORD PATTERN CHECK?
2937 010450 001401      BEQ      65$      ;
2938 010452 104040      ERROR    40      ; PATTERN DOESNT MATCH
2939
2940 010454      65$:
2941 010454 000404      BR      TST135   ;;
2942
2943 010456 125252 125252 125252  ADDR5: .WORD  ALTN,ALTN,ALTN,ALTN ; TEST PATTERN
2944 010464 125252
2945
2946
2947
2948
2949
2950
2951
2952
2953
2954
2955
2956
2957
2958
2959
2960
2961
2962
2963
2964
2965
2966
2967
2968
2969
2970
2971
2972
2973
2974
2975
2976
2977
2978
2979
2980
2981

```

\*\*\*\*\*  
\*TEST 135 TEST OF LOAD-FSRC MODE-5, STORE-FDST MODE-7(PC), D MODE  
\*\*\*\*\*

```

TST135: SCOPE
LDFPS  #047600      ; D MODE FPS
MOV    ADDR6+0,$REG0 ; MOVE PATTERN
MOV    ADDR6+2,$REG1 ;
MOV    ADDR6+4,$REG2 ;
MOV    ADDR6+6,$REG3 ;
;
MOV    #SREG0,$REG10 ; ADDR(DATA)
MOV    #SREG10+2,R1  ; ADDR(ADDR(DATA)+2)
LDD    2-(R1),AC1    ; M5-R1
;
MOV    #SREG4,$REG11 ; ADDR(DEST)
STD    AC1,2SREG11  ; M7-R7
;
CMP    R1,#SREG10    ; WAS FSRC ADDR REG DECRE RIGHT AMOUNT?
BEQ    66$          ;
ERROR  41          ; NOT EQUAL, SIGNAL ERROR
66$:
CMP    $REG0,$REG4  ; 1ST WORD PATTERN CHECK?
BNE    64$          ;
CMP    $REG1,$REG5  ; 2ND WORD PATTERN CHECK?
BNE    64$          ;
CMP    $REG2,$REG6  ; 3RD WORD PATTERN CHECK?
BNE    64$          ;
CMP    $REG3,$REG7  ; 4TH WORD PATTERN CHECK?
BEQ    65$          ;
ERROR  42          ; PATTERN DOESNT MATCH
65$:
BR      TST136     ;;

```

# K05

FPU BASIC INSTR TESTS MACY11 27(1006) 09-FEB-77 09:48 PAGE 64  
 DQFPAA.P11 09-FEB-77 09:42 T135 TEST OF LOAD-FSRC MODE-5, STORE-FDST MODE-7(PC), D MODE

```

2982 010626 007417 007417 007417 ADRD6: .WORD ALT4P,ALT4P,ALT4P,ALT4P ; TEST PATTERN
2983 010634 007417
2984
2985
2986
2987
2988
2989 010636 000004
2990 010640 170127 047600
2991 010644 016767 000130 170316
2992 010652 016767 000124 170312
2993 010660 016767 000120 170306
2994 010666 016767 000114 170302
2995
2996 010674 016767 170270 000002
2997 010702 172627
2998 010704 000000
2999 010706 000403
3000 010710 104120
3001 010712 104121
3002 010714 104122
3003
3004 010716 012767 001200 170264 25: MOV $SREG4,$SREG10 ; ADDR(DEST)
3005 010724 012700 001152 MOV $SREG10-36,$R0 ; ADDR(ADDR(DEST))-36
3006 010730 174270 000036 STD AC2,$36(R0) ; M7-R0
3007
3008 010734 026767 170230 170236 CMP $SREG0,$SREG4 ; 1ST WORD PATTERN CHECK?
3009 010742 001014 BNE 64$ ;
3010 010744 026767 170222 170230 CMP $SREG1,$SREG5 ; 2ND WORD PATTERN CHECK?
3011 010752 001010 BNE 64$ ;
3012 010754 026767 170214 170222 CMP $SREG2,$SREG6 ; 3RD WORD PATTERN CHECK?
3013 010762 001004 BNE 64$ ;
3014 010764 026767 170206 170214 CMP $SREG3,$SREG7 ; 4TH WORD PATTERN CHECK?
3015 010772 001401 BEQ 65$ ;
3016 010774 104043 64$: ERROR 43 ; PATTERN DOESNT MATCH
3017
3018 010776 65$:
3019 010776 000404 BR TST137 ;;
3020
3021 011000 104117 000000 000000 ADRD7: .WORD ERROR117,0,0,0 ; TEST PATTERN
3022 011006 000000
3023
3024
3025
3026
3027
3028 011010 000004
3029 011012 170127 047600
3030 011016 016767 000112 170144
3031 011024 016767 000106 170140
3032 011032 016767 000102 170134
3033 011040 016767 000076 170130
3034
3035 011046 012767 001170 170134 MOV $SREG0,$SREG10 ; ADDR(DATA)
3036 011054 172777 170130 LDD $SREG10,AC3 ; M7-R7
3037

```

```

*****
; *TEST 136 TEST OF LOAD-FSRC MODE-2(PC), STORE-FDST MODE-7, D MODE
*****

```

```

TST136: SCOPE
LDFPS #047600 ; D MODE FPS
MOV ADRD7+0,$SREG0 ; MOVE PATTERN
MOV ADRD7+2,$SREG1
MOV ADRD7+4,$SREG2
MOV ADRD7+6,$SREG3
MOV $SREG0,1$ ; PUT DATA
LDD (PC)+,AC2 ; M2-R7
1$: .WORD 0 ; DATA
BR 2$ ; THIS SHOULD BE NEXT INSTR AFTER FPP PC MODE 2
ERROR 120 ; NOT HERE (+4 INCRE)
ERROR 121 ; OR HERE (+6 INCRE)
ERROR 122 ; OR HERE (+10 INCRE)
2$: MOV $SREG4,$SREG10 ; ADDR(DEST)
MOV $SREG10-36,$R0 ; ADDR(ADDR(DEST))-36
STD AC2,$36(R0) ; M7-R0
3008 010734 026767 170230 170236 CMP $SREG0,$SREG4 ; 1ST WORD PATTERN CHECK?
3009 010742 001014 BNE 64$ ;
3010 010744 026767 170222 170230 CMP $SREG1,$SREG5 ; 2ND WORD PATTERN CHECK?
3011 010752 001010 BNE 64$ ;
3012 010754 026767 170214 170222 CMP $SREG2,$SREG6 ; 3RD WORD PATTERN CHECK?
3013 010762 001004 BNE 64$ ;
3014 010764 026767 170206 170214 CMP $SREG3,$SREG7 ; 4TH WORD PATTERN CHECK?
3015 010772 001401 BEQ 65$ ;
3016 010774 104043 64$: ERROR 43 ; PATTERN DOESNT MATCH
65$:
BR TST137 ;;

```

```

*****
; *TEST 137 TEST OF LOAD-FSRC MODE-7(PC), STORE-FDST MODE-6, D MODE
*****

```

```

TST137: SCOPE
LDFPS #047600 ; D MODE FPS
MOV ADRD10+0,$SREG0 ; MOVE PATTERN
MOV ADRD10+2,$SREG1
MOV ADRD10+4,$SREG2
MOV ADRD10+6,$SREG3
MOV $SREG0,$SREG10 ; ADDR(DATA)
LDD $SREG10,AC3 ; M7-R7

```

```

3038 011060 012704 001212      MOV      #SREG4+12,R4      ; ADDR(DEST)+12
3039 011064 174364 177766      STD      AC3,-12(R4)      ; M6-R4
3040
3041 011070 026767 170074 170102  CMP      $REG0,$REG4      ; 1ST WORD PATTERN CHECK?
3042 011076 001014                BNE      64$              ;
3043 011100 026767 170066 170074  CMP      $REG1,$REG5      ; 2ND WORD PATTERN CHECK?
3044 011106 001010                BNE      64$              ;
3045 011110 026767 170060 170066  CMP      $REG2,$REG6      ; 3RD WORD PATTERN CHECK?
3046 011116 001004                BNE      64$              ;
3047 011120 026767 170052 170060  CMP      $REG3,$REG7      ; 4TH WORD PATTERN CHECK?
3048 011126 001401                BEQ      65$              ;
3049 011130 104044                64$:  ERROR      44      ; PATTERN DOESNT MATCH
3050
3051 011132                65$:
3052 011132 000404                BR      TST140          ;;
3053
3054 011134 170360 170360 170360  ADDR10: .WORD  ALT4N,ALT4N,ALT4N,ALT4N ; TEST PATTERN
3055 011142 170360
3056
3057
3058
3059
3060
3061 011144 000004                ;*****
3062 011146 170127 047600                ;*TEST 140 TEST OF LOAD-FSRC MODE-3, STORE-FDST MODE-2(PC), D MODE
3063 011152 016767 000114 170010  LDFPS   #047600          ;*****
3064 011160 016767 000110 170004  MOV      ADDR11+0,$REG0    ; D MODE FPS
3065 011166 016767 000104 170000  MOV      ADDR11+2,$REG1    ; MOVE PATTERN
3066 011174 016767 000100 167774  MOV      ADDR11+4,$REG2    ;
3067
3068 011202 012767 001170 170000  MOV      #SREG0,$REG10     ; ADDR(DATA)
3069 011210 012702 001210                MOV      #SREG10,R2        ; ADDR(ADDR(DATA))
3070 011214 172432                LDD      3(R2)+,AC0        ; M3-R2
3071
3072 011216 012767 104117 000002  MOV      #ERROR117,1$      ; SETUP ERROR CALL FOR WRONG INCREMENT
3073 011224 174027                STD      AC0,(PC)+        ; M2-R7
3074 011226 000000                1$:  .WORD  0              ; DEST
3075 011230 000403                BR      2$                ; THIS SHOULD BE NEXT INSTR AFTER FPP PC MODE 2
3076 011232 104120                ERROR   120              ; NOT HERE (+4 INCRE)
3077 011234 104121                ERROR   121              ; OR HERE (+6 INCRE)
3078 011236 104122                ERROR   122              ; OR HERE (+10 INCRE)
3079 011240 016767 177762 167732 2$:  MOV      1$,SREG4          ; STORE DEST
3080
3081 011246 020227 001212                CMP      R2,#SREG10+2     ; WAS FSRC ADDR REG INCRE BY RIGHT AMOUNT?
3082 011252 001401                BEQ      64$              ;
3083 011254 104045                64$:  ERROR      45        ; NOT EQUAL, SIGNAL ERROR
3084 011256
3085
3086 011256 026767 167706 167714  CMP      $REG0,$REG4      ; WAS 1 WORD STORED OK?
3087 011264 001401                BEQ      65$              ;
3088 011266 104046                65$:  ERROR      46        ; NOT EQUAL, SIGNAL ERROR
3089 011270
3090
3091 011270 000404                BR      TST141          ;;
3092
3093 011272 052525 000000 000000  ADDR11: .WORD  ALTP,0,0,0  ; TEST PATTERN

```



M05

3094 011300 000000

3095

3096

3097

3098

3099

3100 011302 000004

3101 011304 170127 047600

3102 011310 016767 000116 167652

3103 011316 016767 000112 167646

3104 011324 016767 000106 167642

3105 011332 016767 000102 167636

3106

3107 011340 012703 001262

3108 011344 172563 177706

3109

3110 011350 012701 001210

3111 011354 174141

3112

3113 011356 020127 001200

3114 011362 001401

3115 011364 104047

3116 011366

3117

3118 011366 026767 167576 167604

3119 011374 001014

3120 011376 026767 167570 167576

3121 011404 001010

3122 011406 026767 167562 167570

3123 011414 001004

3124 011416 026767 167554 167562

3125 011424 001401

3126 011426 104050

3127

3128 011430

3129 011430 000404

3130

3131 011432 170360 170360 170360

3132 011440 170360

3133

3134

3135

3136

3137

3138 011442 000004

3139 011444 170127 047600

3140 011450 016767 000102 167512

3141 011456 016767 000076 167506

3142 011464 016767 000072 167502

3143 011472 016767 000066 167476

3144

3145 011500 172637 001170

3146

3147 011504 174203

3148 011506 174367 167466

3149

\*\*\*\*\*  
 ;\*TEST 141 TEST OF LOAD-FSRC MODE-6, STORE-FDST MODE-4, D MODE  
 ;\*\*\*\*\*

TST141: SCOPE  
 LDFPS #047600 ; D MODE FPS  
 MOV ADDR12+0,\$REG0 ; MOVE PATTERN  
 MOV ADDR12+2,\$REG1 ;  
 MOV ADDR12+4,\$REG2 ;  
 MOV ADDR12+6,\$REG3 ;  
 MOV #\$REG0+72,R3 ; ADDR(DATA)+72  
 LDD -72(R3),AC1 ; M6-R3  
 MOV #\$REG4+10,R1 ; ADDR(DEST+10)  
 STD AC1,-(R1) ; M4-R1  
 CMP R1,\$REG4 ; WAS FDST ADDR REG DECRE RIGHT AMOUNT?  
 BEQ 66\$ ;  
 ERROR 47 ; NOT EQUAL, SIGNAL ERROR  
 66\$:  
 CMP \$REG0,\$REG4 ; 1ST WORD PATTERN CHECK?  
 BNE 64\$ ;  
 CMP \$REG1,\$REG5 ; 2ND WORD PATTERN CHECK?  
 BNE 64\$ ;  
 CMP \$REG2,\$REG6 ; 3RD WORD PATTERN CHECK?  
 BNE 64\$ ;  
 CMP \$REG3,\$REG7 ; 4TH WORD PATTERN CHECK?  
 BEQ 65\$ ;  
 ERROR 50 ; PATTERN DOESNT MATCH  
 64\$:  
 65\$:  
 BR TST142 ;  
 ADDR12: .WORD ALT4N,ALT4N,ALT4N,ALT4N ; TEST PATTERN

\*\*\*\*\*  
 ;\*TEST 142 TEST OF LOAD-FSRC MODE-3(PC), STORE-FDST MODE-0, D MODE  
 ;\*\*\*\*\*

TST142: SCOPE  
 LDFPS #047600 ; D MODE FPS  
 MOV ADDR13+0,\$REG0 ; MOVE PATTERN  
 MOV ADDR13+2,\$REG1 ;  
 MOV ADDR13+4,\$REG2 ;  
 MOV ADDR13+6,\$REG3 ;  
 LDD 2\*\$REG0,AC2 ; M3-R7  
 STD AC2,AC3 ; M0-R3  
 STD AC3,\$REG4 ; M6-R7 WORKS

N05

FPU BASIC INSTR TESTS MACY11 27(1006) 09-FEB-77 09:48 PAGE 67  
 DCFPAA.P11 09-FEB-77 09:42 T142 TEST OF LOAD-FSRC MODE-3(PC), STORE-FDST MODE-0, D MODE

3150	011512	026767	167452	167460	CMP	\$REG0,\$REG4	:	1ST WORD PATTERN CHECK?
3151	011520	001014			BNE	64\$	:	
3152	011522	026767	167444	167452	CMP	\$REG1,\$REG5	:	2ND WORD PATTERN CHECK?
3153	011530	001010			BNE	64\$	:	
3154	011532	026767	167436	167444	CMP	\$REG2,\$REG6	:	3RD WORD PATTERN CHECK?
3155	011540	001004			BNE	64\$	:	
3156	011542	026767	167430	167436	CMP	\$REG3,\$REG7	:	4TH WORD PATTERN CHECK?
3157	011550	001401			BEQ	65\$	:	
3158	011552	104051			64\$:	ERROR 51	:	PATTERN DOESNT MATCH
3159								
3160	011554				65\$:			
3161	011554	000404			BR	TST143	::	
3162								
3163	011556	125252	125252	125252	ADDR13:	.WORD ALTN,ALTN,ALTN,ALTN	:	TEST PATTERN
3164	011564	125252						
3165								
3166								
3167								
3168								
3169								
3170	011566	000004			TST143:	SCOPE		
3171	011570	170127	047600		LDFPS	#047600	:	D MODE FPS
3172	011574	016767	000122	167366	MOV	ADDR14+0,\$REG0	:	MOVE PATTERN
3173	011602	016767	000116	167362	MOV	ADDR14+2,\$REG1	:	
3174	011610	016767	000112	167356	MOV	ADDR14+4,\$REG2	:	
3175	011616	016767	000106	167352	MOV	ADDR14+6,\$REG3	:	
3176								
3177	011624	172467	167340		LDD	\$REG0,AC0	:	M6-R7 WORKS
3178	011630	172500			LDD	AC0,AC1	:	M0-R0
3179								
3180	011632	012767	001200	167350	MOV	#\$REG4,\$REG10	:	ADDR(DEST)
3181	011640	012702	001212		MOV	#\$REG10+2,R2	:	ADDR(ADDR(DEST)+2)
3182	011644	174152			STD	AC1,@-(R2)	:	M5-R2
3183								
3184	011646	020227	001210		CMP	R2,\$REG10	:	WAS FDST ADDR REG DECRE RIGHT AMOUNT?
3185	011652	001401			BEQ	66\$	:	
3186	011654	104052			66\$:	ERROR 52	:	NOT EQUAL, SIGNAL ERROR
3187	011656							
3188								
3189	011656	026767	167306	167314	CMP	\$REG0,\$REG4	:	1ST WORD PATTERN CHECK?
3190	011664	001014			BNE	64\$	:	
3191	011666	026767	167300	167306	CMP	\$REG1,\$REG5	:	2ND WORD PATTERN CHECK?
3192	011674	001010			BNE	64\$	:	
3193	011676	026767	167272	167300	CMP	\$REG2,\$REG6	:	3RD WORD PATTERN CHECK?
3194	011704	001004			BNE	64\$	:	
3195	011706	026767	167264	167272	CMP	\$REG3,\$REG7	:	4TH WORD PATTERN CHECK?
3196	011714	001401			BEQ	65\$	:	
3197	011716	104046			64\$:	ERROR 46	:	PATTERN DOESNT MATCH
3198								
3199	011720				65\$:			
3200	011720	000404			BR	TST144	::	
3201								
3202	011722	125252	125252	125252	ADDR14:	.WORD ALTN,ALTN,ALTN,ALTN	:	TEST PATTERN
3203	011730	125252						
3204								
3205								

```

3206
3207
3208
3209 011732 000004
3210 011734 170127 047400
3211 011740 016767 000060 167222
3212 011746 016767 000054 167216
3213
3214 011754 012702 001170
3215 011760 172412
3216
3217 011762 012705 001200
3218 011766 174045
3219
3220 011770 020527 001174
3221 011774 001401
3222 011776 104053
3223 012000
3224
3225 012000 026767 167164 167166
3226 012006 001004
3227 012010 026767 167156 167160
3228 012016 001401
3229 012020 104054
3230
3231 012022
3232 012022 000402
3233
3234 012024 170360 170360
3235
3236
3237
3238
3239
3240 012030 000004
3241 012032 170127 047400
3242 012036 016767 000070 167124
3243 012044 016767 000064 167120
3244
3245 012052 012767 001170 167120
3246 012060 012704 001200
3247 012064 172534
3248
3249 012066 012700 001166
3250 012072 174160 000006
3251
3252 012076 020427 001202
3253 012102 001401
3254 012104 104055
3255 012106
3256
3257 012106 026767 167056 167060
3258 012114 001004
3259 012116 026767 167050 167052
3260 012124 001401
3261 012126 104056

```

```

*****
*TEST 144 TEST OF LOAD-FSRC MODE-1, STORE-FDST MODE-4, F MODE
*****
TST144: SCOPE
LDFPS #047400 ; F MODE FPS
MOV ADRF1+0,$REG0 ; MOVE PATTERN
MOV ADRF1+2,$REG1 ;
MOV #SREG0,R2 ; ADDR(DATA)
LDF (R2),AC0 ; M1-R2
MOV #SREG2+4,R5 ; ADDR(DEST+4)
STF AC0,-(R5) ; M4-R5
CMP R5,#SREG2 ; WAS FDST ADDR REG DECRE RIGHT AMOUNT?
BEQ 66$ ;
ERROR 53 ; NOT EQUAL, SIGNAL ERROR
66$:
CMP SREG0,$REG2 ; 1ST WORD PATTERN CHECK?
BNE 64$ ;
CMP SREG1,$REG3 ; 2ND WORD PATTERN CHECK?
BEQ 65$ ;
ERROR 54 ; PATTERN DOESNT MATCH
65$:
BR TST145 ;;
ADRF1: .WORD ALT4N,ALT4N ; TEST PATTERN

```

```

*****
*TEST 145 TEST OF LOAD-FSRC MODE-3, STORE-FDST MODE-6, F MODE
*****
TST145: SCOPE
LDFPS #047400 ; F MODE FPS
MOV ADRF2+0,$REG0 ; MOVE PATTERN
MOV ADRF2+2,$REG1 ;
MOV #SREG0,$REG4 ; ADDR(DATA)
MOV #SREG4,R4 ; ADDR(ADDR(DATA))
LDF @ (R4)+,AC1 ; M3-R4
MOV #SREG2-6,R0 ; ADDR(DEST)-6
STF AC1,6(R0) ; M6-R0
CMP R4,#SREG4+2 ; WAS FSRC ADDR REG INCRE RIGHT AMOUNT?
BEQ 66$ ;
ERROR 55 ; NOT EQUAL, SIGNAL ERROR
66$:
CMP SREG0,$REG2 ; 1ST WORD PATTERN CHECK?
BNE 64$ ;
CMP SREG1,$REG3 ; 2ND WORD PATTERN CHECK?
BEQ 65$ ;
ERROR 56 ; PATTERN DOESNT MATCH

```

```

3262
3263 012130
3264 012130 000402
3265
3266 012132 052525 052525
3267
3268
3269
3270
3271
3272 012136 000004
3273 012140 170127 047400
3274 012144 016767 000104 167016
3275 012152 016767 000100 167012
3276
3277 012160 012767 001170 167012
3278 012166 012700 001202
3279 012172 172650
3280
3281 012174 012767 001174 167000
3282 012202 012702 001202
3283 012206 174232
3284
3285 012210 020027 001200
3286 012214 001401
3287 012216 104057
3288 012220
3289
3290 012220 020227 001204
3291 012224 001401
3292 012226 104060
3293 012230
3294
3295 012230 026767 166734 166736
3296 012236 001004
3297 012240 026767 166726 166730
3298 012246 001401
3299 012250 104061
3300
3301 012252
3302 012252 000402
3303
3304 012254 007417 007417
3305
3306
3307
3308
3309
3310 012260 000004
3311 012262 170127 047400
3312 012266 016767 000060 166674
3313 012274 016767 000054 166670
3314
3315 012302 012767 001170 166670
3316 012310 012702 001162
3317 012314 172772 000016

```

```

65$: BR TST146 ;;
ADRF2: .WORD ALTP,ALTP ; TEST PATTERN

;*****
; *TEST 146 TEST OF LOAD-FSRC MODE-5, STORE-FDST MODE-3, F MODE
;*****
TST146: SCOPE
LDFPS #047400 ; F MODE FPS
MOV ADRF3+0,$REG0 ; MOVE PATTERN
MOV ADRF3+2,$REG1 ;
MOV #SREG0,$REG4 ; ADDR(DATA)
MOV #SREG4+2,R0 ; ADDR(ADDR(DATA)+2)
LDF @-(R0),AC2 ; M5-R0
MOV #SREG2,$REG5 ; ADDR(DEST)
MOV #SREG5,R2 ; ADDR(ADDR(DEST))
STF AC2,@(R2)+ ; M3-R2
CMP R0,#SREG4 ; WAS FSRC ADDR REG DECREASE RIGHT AMOUNT?
BEQ 66$ ;
ERROR 57 ; NOT EQUAL, SIGNAL ERROR
66$:
CMP R2,#SREG5+2 ; WAS FDST ADDR REG INCREASE RIGHT AMOUNT?
BEQ 67$ ;
ERROR 60 ; NOT EQUAL, SIGNAL ERROR
67$:
CMP $REG0,$REG2 ; 1ST WORD PATTERN CHECK?
BNE 64$ ;
CMP $REG1,$REG3 ; 2ND WORD PATTERN CHECK?
BEQ 65$ ;
ERROR 61 ; PATTERN DOESNT MATCH
64$:
65$: BR TST147 ;;
ADRF3: .WORD ALT4P,ALT4P ; TEST PATTERN

;*****
; *TEST 147 TEST OF LOAD-FSRC MODE-7, STORE-FDST MODE-1, F MODE
;*****
TST147: SCOPE
LDFPS #047400 ; F MODE FPS
MOV ADRF4+0,$REG0 ; MOVE PATTERN
MOV ADRF4+2,$REG1 ;
MOV #SREG0,$REG4 ; ADDR(DATA)
MOV #SREG4-16,R2 ; ADDR(ADDR(DATA))-16
LDF @16(R2),AC3 ; M7-R2

```

```

3318
3319 012320 012701 001174      MOV    #SREG2,R1      ; ADDR(DEST)
3320 012324 174311      STF    AC3,(R1)      ; M1-R1
3321
3322 012326 026767 166636 166640      CMP    $REG0,$REG2   ; 1ST WORD PATTERN CHECK?
3323 012334 001004      BNE    64$           ;
3324 012336 026767 166630 166632      CMP    $REG1,$REG3   ; 2ND WORD PATTERN CHECK?
3325 012344 001401      BEQ    65$           ;
3326 012346 104062      64$:  ERROR    62      ; PATTERN DOESNT MATCH
3327
3328 012350      65$:
3329 012350 000402      BR     TST150        ;;
3330
3331 012352 125252 125252      ADRF4: .WORD  ALTN,ALTN ; TEST PATTERN
3332
3333
3334
3335      ;*****
3336      ;*TEST 150      TEST OF LOAD-FSRC MODE-3(PC), STORE-FDST MODE-2, F MODE
3337      ;*****
3337 012356 000004      TST150: SCOPE
3338 012360 170127 047400      LDFPS  #047400      ; F MODE FPS
3339 012364 016767 000056 166576      MOV    ADRF5+0,$REG0 ; MOVE PATTERN
3340 012372 016767 000052 166572      MOV    ADRF5+2,$REG1 ;
3341
3342 012400 172437 001170      LDF    2#$REG0,AC0   ; M3-R7
3343
3344 012404 012703 001174      MOV    #SREG2,R3     ; ADDR(DEST)
3345 012410 174023      STF    AC0,(R3)+     ; M2-R3
3346
3347 012412 020327 001200      CMP    R3,#SREG2+4   ; WAS FDST ADDR REG INCRE RIGHT AMOUNT?
3348 012416 001401      BEQ    66$           ;
3349 012420 104063      66$:  ERROR    63      ; NOT EQUAL, SIGNAL ERROR
3350 012422
3351
3352 012422 026767 166542 166544      CMP    $REG0,$REG2   ; 1ST WORD PATTERN CHECK?
3353 012430 001004      BNE    64$           ;
3354 012432 026767 166534 166536      CMP    $REG1,$REG3   ; 2ND WORD PATTERN CHECK?
3355 012440 001401      BEQ    65$           ;
3356 012442 104064      64$:  ERROR    64      ; PATTERN DOESNT MATCH
3357
3358 012444      65$:
3359 012444 000402      BR     TST151        ;;
3360
3361 012446 007417 007417      ADRF5: .WORD  ALT4P,ALT4P ; TEST PATTERN
3362
3363
3364
3365      ;*****
3366      ;*TEST 151      TEST OF LOAD-FSRC MODE-7(PC), STORE-FDST MODE-7, F MODE
3367      ;*****
3367 012452 000004      TST151: SCOPE
3368 012454 170127 047400      LDFPS  #047400      ; F MODE FPS
3369 012460 016767 000064 166502      MOV    ADRF6+0,$REG0 ; MOVE PATTERN
3370 012466 016767 000060 166476      MOV    ADRF6+2,$REG1 ;
3371
3372 012474 012767 001170 166476      MOV    #SREG0,$REG4  ; ADDR(DATA)
3373 012502 172577 166472      LDF    2#$REG4,AC1  ; M7-R7

```

E06

FPU BASIC INSTR TESTS MACY11 27(1006) 09-FEB-77 09:48 PAGE 71  
 DQFPAA.P11 09-FEB-77 09:42 T151 TEST OF LOAD-FSRC MODE-7(PC), STORE-FDST MODE-7, F MODE

```

3374
3375 012506 012767 001174 166466      MOV    #SREG2,SREG5      ; ADDR(DEST)
3376 012514 012705 001226              MOV    #SREG5+24,R5      ; ADDR(ADDR(DEST))+24
3377 012520 174175 177754      STF    AC1,@-24(R5)      ; M7-R5
3378
3379 012524 026767 166440 166442      CMP    $REG0,$REG2      ; 1ST WORD PATTERN CHECK?
3380 012532 001004              BNE    64$              ;
3381 012534 026767 166432 166434      CMP    $REG1,$REG3      ; 2ND WORD PATTERN CHECK?
3382 012542 001401              BEQ    65$              ;
3383 012544 104065              64$:  ERROR    65$      ; PATTERN DOESNT MATCH
3384
3385 012546              65$:
3386 012546 000402              BR     TST152           ;;
3387
3388 012550 125252 125252      ADRF6: .WORD  ALTN,ALTN  ; TEST PATTERN
3389
3390
3391
3392
3393
3394 012554 000004      ;:*****
3395 012556 170127 047400      ;: *TEST 152      TEST OF LOAD-FSRC MODE-2, STORE-FDST MODE-3(PC), F MODE
3396 012562 016767 000056 166400      ;:*****
3397 012570 016767 000052 166374      †TST152: SCOPE
3398
3399 012576 012703 001170      LDFPS  #047400          ; F MODE FPS
3400 012602 172623      MOV    ADRF7+0,$REG0    ; MOVE PATTERN
3401
3402 012604 174237 001174      MOV    ADRF7+2,$REG1    ;
3403
3404 012610 020327 001174      MOV    #SREG0,R3        ; ADDR(DATA)
3405 012614 001401      LDF    (R3)+,AC2        ; M2-R3
3406 012616 104066              STF    AC2,@#SREG2      ; M3-R7
3407 012620              64$:
3408
3409 012620 026767 166344 166346      CMP    $REG0,$REG2      ; 1ST WORD PATTERN CHECK?
3410 012626 001004              BNE    64$              ;
3411 012630 026767 166336 166340      CMP    $REG1,$REG3      ; 2ND WORD PATTERN CHECK?
3412 012636 001401              BEQ    65$              ;
3413 012640 104064              64$:  ERROR    64$      ; PATTERN DOESNT MATCH
3414
3415 012642              65$:
3416 012642 000402              BR     TST153           ;;
3417
3418 012644 170360 170360      ADRF7: .WORD  ALT4N,ALT4N ; TEST PATTERN
3419
3420
3421
3422
3423
3424 012650 000004      ;:*****
3425 012652 170127 047400      ;: *TEST 153      TEST OF LOAD-FSRC MODE-4, STORE-FDST MODE-5, F MODE
3426 012656 016767 000076 166304      ;:*****
3427 012664 016767 000072 166300      †TST153: SCOPE
3428
3429 012672 012705 001174      LDFPS  #047400          ; F MODE FPS
3429 012672 012705 001174      MOV    ADRF10+0,$REG0   ; MOVE PATTERN
3429 012672 012705 001174      MOV    ADRF10+2,$REG1   ;
3429 012672 012705 001174      MOV    #SREG0+4,R5      ; ADDR(DATA+4)
  
```

FPU BASIC INSTR TESTS MACY11 27(1006) 09-FEB-77 09:48 PAGE 72  
 DQFPAA.P11 09-FEB-77 09:42 T153 TEST OF LOAD-FSRC MODE-4, STORE-FDST MODE-5, F MODE

```

3430 012676 172745          LDF      -(R5),AC3          ; M4-R5
3431
3432 012700 012767 001174 166272  MOV      #$REG2,$REG4      ; ADDR(DEST)
3433 012706 012704 001202          MOV      #$REG4+2,R4       ; ADDR(ADDR(DEST)+2)
3434 012712 174354          STF      AC3,2-(R4)        ; M5-R4
3435
3436 012714 020527 001170          CMP      R5,$REG0          ; WAS FSRC ADDR REG DECRE RIGHT AMOUNT?
3437 012720 001401          BEQ     66$                ;
3438 012722 104067          ERROR   67                ; NOT EQUAL, SIGNAL ERROR
3439 012724          66$:
3440
3441 012724 020427 001200          CMP      R4,$REG4          ; WAS FDST ADDR REG DECRE RIGHT AMOUNT?
3442 012730 001401          BEQ     67$                ;
3443 012732 104070          ERROR   70                ; NOT EQUAL, SIGNAL ERROR
3444 012734          67$:
3445
3446 012734 026767 166230 166232  CMP      $REG0,$REG2       ; 1ST WORD PATTERN CHECK?
3447 012742 001004          BNE     64$                ;
3448 012744 026767 166222 166224  CMP      $REG1,$REG3       ; 2ND WORD PATTERN CHECK?
3449 012752 001401          BEQ     65$                ;
3450 012754 104071          ERROR   71                ; PATTERN DOESNT MATCH
3451
3452 012756          65$:
3453 012756 000402          BR      TST154            ;;
3454
3455 012760 052525 052525  ADRF10: .WORD  ALTP,ALTP    ; TEST PATTERN
3456
3457
3458 ;:*****
3459 ;:TEST 154 TEST OF LOAD-FSRC MODE-2(PC), STORE-FDST MODE-7(PC), F MODE
3460 ;:*****
3461 012764 000004          †TST154: SCOPE
3462 012766 170127 047400          LDFPS   #047400           ; F MODE FPS
3463 012772 016767 000070 166170  MOV      ADRF11+0,$REG0    ; MOVE PATTERN
3464 013000 016767 000064 166164  MOV      ADRF11+2,$REG1    ;
3465
3466 013006 016767 166156 000002  MOV      $REG0,1$         ; PUT DATA
3467 013014 172427          LDF     (PC)+,AC0         ; M2-R7
3468 013016 000000          .WORD  0                  ; DATA
3469 013020 000403          BR      2$                ; THIS SHOULD BE NEXT INSTR AFTER FPP PC MODE 2
3470 013022 104120          ERROR  120                ; NOT HERE (+4 INCRE)
3471 013024 104121          ERROR  121                ; OR HERE (+6 INCRE)
3472 013026 104122          ERROR  122                ; OR HERE (+10 INCRE)
3473
3474 013030 012767 001174 166142 2$:  MOV      #$REG2,$REG4      ; ADDR(DEST)
3475 013036 174077 166136          STF     AC0,2$REG4        ; M7-R7
3476
3477 013042 026767 166122 166124  CMP      $REG0,$REG2       ; 1ST WORD PATTERN CHECK?
3478 013050 001004          BNE     64$                ;
3479 013052 026767 166114 166116  CMP      $REG1,$REG3       ; 2ND WORD PATTERN CHECK?
3480 013060 001401          BEQ     65$                ;
3481 013062 104072          ERROR   72                ; PATTERN DOESNT MATCH
3482
3483 013064          65$:
3484 013064 000402          BR      TST155            ;;
3485

```

```

3486 013066 104117 000000 ADRF11: .WORD ERROR117,0 ; TEST PATTERN
3487
3488
3489
3490
3491
3492 013072 000004
3493 013074 170127 047400
3494 013100 016767 000044 166062
3495 013106 016767 000040 166056
3496
3497 013114 172567 166050 LDF $REG0,AC1 ; M6-R7
3498
3499 013120 174167 166050 STF AC1,$REG2 ; M6-R7
3500
3501 013124 026767 166040 166042 CMP $REG0,$REG2 ; 1ST WORD PATTERN CHECK?
3502 013132 001004 BNE 64$ ;
3503 013134 026767 166032 166034 CMP $REG1,$REG3 ; 2ND WORD PATTERN CHECK?
3504 013142 001401 BEQ 65$ ;
3505 013144 104073 64$: ERROR 73 ; PATTERN DOESNT MATCH
3506
3507 013146 65$:
3508 013146 000402 BR TST156 ;;
3509
3510 013150 052525 052525 ADRF12: .WORD ALTP,ALTP ; TEST PATTERN
3511
3512
3513
3514
3515
3516 013154 000004
3517 013156 170127 047400
3518 013162 016767 000062 166000
3519 013170 016767 000056 165774
3520
3521 013176 172667 165766 LDF $REG0,AC2 ; M6-R7 WORKS
3522 013202 172702 LDF AC2,AC3 ; M3-R2
3523
3524 013204 012767 104117 000002 MOV #ERROR117,1$ ; SETUP ERROR CALL FOR WRONG INCREMENT
3525 013212 174327 STF AC3,(PC)+ ; M2-R7
3526 013214 000000 1$: .WORD 0 ; DEST
3527 013216 000403 BR 2$ ; THIS SHOULD BE NEXT INSTR AFTER FPP PC MODE 2
3528 013220 104120 ERROR 120 ; NOT HERE (+4 INCRE)
3529 013222 104121 ERROR 121 ; OR HERE (+6 INCRE)
3530 013224 104122 ERROR 122 ; OR HERE (+10 INCRE)
3531 013226 016767 177762 165740 2$: MOV 1$,$REG2 ; GET DEST
3532
3533 013234 026767 165730 165732 CMP $REG0,$REG2 ; WAS 1 WORD STORED OK?
3534 013242 001401 BEQ 64$ ;
3535 013244 104073 64$: ERROR 73 ; NOT EQUAL, SIGNAL ERROR
3536 013246
3537
3538 013246 000402 BR TST157 ;;
3539
3540 013250 007417 000000 ADRF13: .WORD ALT4P,0 ; TEST PATTERN
3541

```



3542  
3543  
3544  
3545  
3546  
3547  
3548  
3549  
3550  
3551  
3552  
3553  
3554  
3555  
3556  
3557  
3558  
3559  
3560  
3561  
3562  
3563  
3564  
3565  
3566  
3567  
3568

013254 000004  
013256 170127 047400  
013262 016767 000052 165700  
013270 016767 000046 165674  
  
013276 012701 001200  
013302 172461 177770  
  
013306 174001  
013310 174167 165660  
  
013314 026767 165650 165652  
013322 001004  
013324 026767 165642 165644  
013332 001401  
013334 104074  
  
013336  
013336 000402  
  
013340 007417 007417

```
*****  
; TEST 157 TEST OF LOAD-FSRC MODE-6, STORE-FDST MODE-0, F MODE  
*****  
†ST157: SCOPE  
LDFPS #047400 ; F MODE FPS  
MOV ADRF14+0,$REG0 ; MOVE PATTERN  
MOV ADRF14+2,$REG1 ;  
  
MOV #SREG0+10,R1 ; ADDR(DATA)+10  
LDF -10(R1),ACO ; M6-R1  
  
STF ACO,AC1 ; M0-R1  
STF AC1,$REG2 ; M6-R7 WORKS  
  
CMP $REG0,$REG2 ; 1ST WORD PATTERN CHECK?  
BNE 64$ ;  
CMP $REG1,$REG3 ; 2ND WORD PATTERN CHECK?  
BEQ 65$ ;  
ERROR 74 ; PATTERN DOESNT MATCH  
  
64$:  
65$:  
BR TST160 ;;  
  
ADRF14: .WORD ALT4P,ALT4P ; TEST PATTERN
```

```

3569
3570
3571
3572
3573
3574
3575
3576
3577 013344 000004
3578 013346 170127 047400
3579 013352 005037 001170
3580 013356 012737 000005 001172
3581 013364 172437 001170
3582 013370 174005
3583 013372 005337 001172
3584 013376 172437 001170
3585 013402 174004
3586 013404 005337 001172
3587 013410 172737 001170
3588 013414 005337 001172
3589 013420 172637 001170
3590 013424 005337 001172
3591 013430 172537 001170
3592 013434 005337 001172
3593 013440 172437 001170
3594 013444 005037 001170
3595 013450 005037 001172
3596 013454 174037 001174
3597 013460 005737 001174
3598 013464 001004
3599 013466 023737 001172 001176
3600 013474 001401
3601 013476 104075 1S:
3602 013500 005237 001172 11S:
3603 013504 174137 001174
3604 013510 005737 001174
3605 013514 001004
3606 013516 023737 001172 001176
3607 013524 001401
3608 013526 104075 2S:
3609 013530 005237 001172 21S:
3610 013534 174237 001174
3611 013540 005737 001174
3612 013544 001004
3613 013546 023737 001172 001176
3614 013554 001401
3615 013556 104075 3S:
3616 013560 005237 001172 31S:
3617 013564 174337 001174
3618 013570 005737 001174
3619 013574 001004
3620 013576 023737 001172 001176
3621 013604 001401
3622 013606 104075 4S:
3623 013610 005237 001172 41S:
3624 013614 172404

```

```

;*****
; .ENABL AMA ; BEGIN ASSEMBLING RELATIVE REFERENCES AS ABSOLUTE
;*****
;*****
; *TEST 160 TEST ALL FP-ACCUM ARE THERE
;*****
TST160: SCOPE
LDFPS #047400 ; INITIAL FPS (SINGLE FLOAT, INTEGER)
CLR $REG0 ; SET AC0 = 0,
MOV #5, $REG1 ; AC1 = 1,
LDF $REG0, AC0 ; AC2 = 2,
STF AC0, AC5 ; AC3 = 3,
DEC $REG1 ; AC4 = 4,
LDF $REG0, AC0 ; AC5 = 5
STF AC0, AC4 ; ...
DEC $REG1 ; ...
LDF $REG0, AC3 ; ...
DEC $REG1 ; ...
LDF $REG0, AC2 ; ...
DEC $REG1 ; ...
LDF $REG0, AC1 ; ...
DEC $REG1 ; ...
LDF $REG0, AC0 ; ...
CLR $REG0 ; NOW TEST THE ABOVE IS TRUE:
CLR $REG1 ; DOES AC0 = 0 ?
STF AC0, $REG2 ; ...
TST $REG2 ; ...
BNE 1S ; ...
CMP $REG1, $REG3 ; ...
BEQ 11S ; ...
INC $REG1 ; NO, ERROR
STF AC1, $REG2 ; YES, INC FOR NEXT TEST
TST $REG2 ; DOES AC1 = 1 ?
BNE 2S ; ...
CMP $REG1, $REG3 ; ...
BEQ 21S ; ...
INC $REG1 ; NO, ERROR
STF AC2, $REG2 ; YES, INC FOR NEXT TEST
TST $REG2 ; DOES AC2 = 2 ?
BNE 3S ; ...
CMP $REG1, $REG3 ; ...
BEQ 31S ; ...
INC $REG1 ; NO, ERROR
STF AC3, $REG2 ; YES, INC FOR NEXT TEST
TST $REG2 ; DOES AC3 = 3 ?
BNE 4S ; ...
CMP $REG1, $REG3 ; ...
BEQ 41S ; ...
INC $REG1 ; NO, ERROR
LDF AC4, AC0 ; YES, INC FOR NEXT TEST
; DOES AC4 = 4 ?

```

FPU BASIC INSTR TESTS MACY11 27(1006) 09-FEB-77 09:48 PAGE 76  
DQFPA.P11 09-FEB-77 09:42 T160 TEST ALL FP-ACCUM ARE THERE

3625	013616	174037	001174		STF	AC0,\$REG2	:	...
3626	013622	005737	001174		TST	\$REG2	:	...
3627	013626	001004			BNE	5\$	:	...
3628	013630	023737	001172	001176	CMP	\$REG1,\$REG3	:	...
3629	013636	001401			BEQ	51\$	:	...
3630	013640	104075			5\$:	ERROR	:	NO, ERROR
3631	013642	005237	001172		51\$:	INC	:	YES, INC FOR NEXT TEST
3632	013646	172405			LDF	AC5,AC0	:	DOES AC5 = 5 ?
3633	013650	174037	001174		STF	AC0,\$REG2	:	...
3634	013654	005737	001174		TST	\$REG2	:	...
3635	013660	001004			BNE	6\$	:	...
3636	013662	023737	001172	001176	CMP	\$REG1,\$REG3	:	...
3637	013670	001401			BEQ	61\$	:	...
3638	013672	104075			6\$:	ERROR	:	NO, ERROR
3639	013674				61\$:		:	YES, GO FOR NEXT TEST

```

3640
3641
3642
3643
3644 013674 000004
3645 013676 012705 013710
3646 013702 004737 021376
3647
3648 013706 000407
3649
3650 013710
3651 013710 000000 000000
3652 013714 000000 000000
3653 013720 047453 047444
3654 013724 000000
3655
3656
3657
3658
3659
3660
3661 013726 000004
3662 013730 012705 013742
3663 013734 004737 021376
3664
3665 013740 000407
3666
3667 013742
3668 013742 052525 052525
3669 013746 052525 052525
3670 013752 047557 047540
3671 013756 000000
3672
3673
3674
3675
3676
3677
3678 013760 000004
3679 013762 012705 013774
3680 013766 004737 021376
3681
3682 013772 000407
3683
3684 013774
3685 013774 125252 125252
3686 014000 025252 125252
3687 014004 047417 047400
3688 014010 000000
3689
3690
3691
3692
3693
3694
3695 014012 000004

```

```

*****
; *TEST 161 TEST OF ABSF INSTR, DATA SET ABSF-1
; * TRUNCATE MODE, ALL INTERRUPT ENABLES ON
*****
†TST161: SCOPE
MOV #ABSF1,R5 ; PTR TO TEST DATA SET
JSR PC,#ABSF1 ; GO TEST
BR TST162 ;;

ABSF1: ; TEST DATA SET ABSF-1:
.WORD 0,0 ; INITIAL MEM FLOAT NUMBER
.WORD 0,0 ; EXPECTED FLOAT RESULT
.WORD 047453,047444 ; FPS: BEFORE, AFTER
.WORD 000000 ; FEC AFTER ( 0 = N/A )

*****
; *TEST 162 TEST OF ABSF INSTR, DATA SET ABSF-2
; * TRUNCATE MODE, ALL INTERRUPT ENABLES ON
*****
†TST162: SCOPE
MOV #ABSF2,R5 ; PTR TO TEST DATA SET
JSR PC,#ABSF2 ; GO TEST
BR TST163 ;;

ABSF2: ; TEST DATA SET ABSF-2:
.WORD ALTP,ALTP ; INITIAL MEM FLOAT NUMBER
.WORD ALTP,ALTP ; EXPECTED FLOAT RESULT
.WORD 047557,047540 ; FPS: BEFORE, AFTER
.WORD 000000 ; FEC AFTER ( 0 = N/A )

*****
; *TEST 163 TEST OF ABSF INSTR, DATA SET ABSF-3
; * ROUNDING MODE, ALL INTERRUPT ENABLES ON
*****
†TST163: SCOPE
MOV #ABSF3,R5 ; PTR TO TEST DATA SET
JSR PC,#ABSF3 ; GO TEST
BR TST164 ;;

ABSF3: ; TEST DATA SET ABSF-3:
.WORD ALTN,ALTN ; INITIAL MEM FLOAT NUMBER
.WORD 025252,ALTN ; EXPECTED FLOAT RESULT
.WORD 047417,047400 ; FPS: BEFORE, AFTER
.WORD 000000 ; FEC AFTER ( 0 = N/A )

*****
; *TEST 164 TEST OF ABSF INSTR, DATA SET ABSF-4
; * ROUNDING MODE, ALL INTERRUPT ENABLES ON
*****
†TST164: SCOPE

```

3696 014014 012705 014026  
3697 014020 004737 021376  
3698  
3699 014024 000407  
3700  
3701 014026  
3702 014026 177777 177777  
3703 014032 077777 177777  
3704 014036 047517 047500  
3705 014042 000000  
3706  
3707  
3708  
3709  
3710  
3711  
3712 014044 000004  
3713 014046 012705 014060  
3714 014052 004737 021376  
3715  
3716 014056 000407  
3717  
3718 014060  
3719 014060 000200 000000  
3720 014064 000200 000000  
3721 014070 047457 047440  
3722 014074 000000  
3723  
3724  
3725  
3726  
3727  
3728  
3729 014076 000004  
3730 014100 012705 014112  
3731 014104 004737 021376  
3732  
3733 014110 000407  
3734  
3735 014112  
3736 014112 077777 177777  
3737 014116 077777 177777  
3738 014122 047557 047540  
3739 014126 000000  
3740  
3741  
3742  
3743  
3744  
3745  
3746 014130 000004  
3747 014132 012705 014144  
3748 014136 004737 021376  
3749  
3750 014142 000407  
3751

MOV #ABSF4,R5 ; PTR TO TEST DATA SET  
JSR PC,2#ABSFT ; GO TEST  
BR TST165 ;;  
ABSF4: ; TEST DATA SET ABSF-4:  
.WORD LGN,M1 ; INITIAL MEM FLOAT NUMBER  
.WORD LGP,M1 ; EXPECTED FLOAT RESULT  
.WORD 047517,047500 ; FPS: BEFORE, AFTER  
.WORD 000000 ; FEC AFTER ( 0 = N/A )

\*\*\*\*\*  
\*TEST 165 TEST OF ABSF INSTR, DATA SET ABSF-5  
\* TRUNCATE MODE, ALL INTERRUPT ENABLES ON  
\*\*\*\*\*

TST165: SCOPE  
MOV #ABSF5,R5 ; PTR TO TEST DATA SET  
JSR PC,2#ABSFT ; GO TEST  
BR TST166 ;;

ABSF5: ; TEST DATA SET ABSF-5:  
.WORD SMP,0 ; INITIAL MEM FLOAT NUMBER  
.WORD SMP,0 ; EXPECTED FLOAT RESULT  
.WORD 047457,047440 ; FPS: BEFORE, AFTER  
.WORD 000000 ; FEC AFTER ( 0 = N/A )

\*\*\*\*\*  
\*TEST 166 TEST OF ABSF INSTR, DATA SET ABSF-6  
\* TRUNCATE MODE, ALL INTERRUPT ENABLES ON  
\*\*\*\*\*

TST166: SCOPE  
MOV #ABSF6,R5 ; PTR TO TEST DATA SET  
JSR PC,2#ABSFT ; GO TEST  
BR TST167 ;;

ABSF6: ; TEST DATA SET ABSF-6:  
.WORD LGP,M1 ; INITIAL MEM FLOAT NUMBER  
.WORD LGP,M1 ; EXPECTED FLOAT RESULT  
.WORD 047557,047540 ; FPS: BEFORE, AFTER  
.WORD 000000 ; FEC AFTER ( 0 = N/A )

\*\*\*\*\*  
\*TEST 167 TEST OF ABSF INSTR, DATA SET ABSF-7  
\* ROUNDING MODE, ALL INTERRUPT ENABLES ON  
\*\*\*\*\*

TST167: SCOPE  
MOV #ABSF7,R5 ; PTR TO TEST DATA SET  
JSR PC,2#ABSFT ; GO TEST  
BR TST170 ;;

3752 014144  
3753 014144 100200 000000  
3754 014150 000200 000000  
3755 014154 047417 047400  
3756 014160 000000

ABSF7: ; TEST DATA SET ABSF-7:  
.WORD SMN,0 ; INITIAL MEM FLOAT NUMBER  
.WORD SMP,0 ; EXPECTED FLOAT RESULT  
.WORD 047417,047400 ; FPS: BEFORE, AFTER  
.WORD 000000 ; FEC AFTER ( 0 = N/A )

3757  
3758  
3759  
3760  
3761  
3762

\*\*\*\*\*  
; TEST 170 TEST OF ABSF INSTR, DATA SET ABSF-10  
; \* ROUNDING MODE, ALL INTERRUPT ENABLES ON  
\*\*\*\*\*

3763 014162 000004  
3764 014164 012705 014176  
3765 014170 004737 021376  
3766  
3767 014174 000407

TST170: SCOPE  
MOV #ABSF10,R5 ; PTR TO TEST DATA SET  
JSR PC,#ABSF7 ; GO TEST  
  
BR TST171 ;;

3768  
3769 014176  
3770 014176 000177 177777  
3771 014202 000000 000000  
3772 014206 047513 047504  
3773 014212 000000

ABSF10: ; TEST DATA SET ABSF-10:  
.WORD ZXIMP,M1 ; INITIAL MEM FLOAT NUMBER  
.WORD 0,0 ; EXPECTED FLOAT RESULT  
.WORD 047513,047504 ; FPS: BEFORE, AFTER  
.WORD 000000 ; FEC AFTER ( 0 = N/A )

3774  
3775  
3776  
3777  
3778  
3779

\*\*\*\*\*  
; TEST 171 TEST OF ABSF INSTR, DATA SET ABSF-11  
; \* TRUNCATE MODE, ALL INTERRUPT ENABLES ON  
\*\*\*\*\*

3780 014214 000004  
3781 014216 012705 014230  
3782 014222 004737 021376  
3783  
3784 014226 000407

TST171: SCOPE  
MOV #ABSF11,R5 ; PTR TO TEST DATA SET  
JSR PC,#ABSF7 ; GO TEST  
  
BR TST172 ;;

3785  
3786 014230  
3787 014230 100000 000000  
3788 014234 000000 000000  
3789 014240 047453 147444  
3790 014244 100014

ABSF11: ; TEST DATA SET ABSF-11:  
.WORD MO,0 ; INITIAL MEM FLOAT NUMBER  
.WORD 0,0 ; EXPECTED FLOAT RESULT  
.WORD 047453,147444 ; FPS: BEFORE, AFTER  
.WORD 100014 ; FEC AFTER ( 0 = N/A )

3791  
3792  
3793  
3794  
3795  
3796

\*\*\*\*\*  
; TEST 172 TEST OF ABSF INSTR, DATA SET ABSF-12  
; \* TRUNCATE MODE, ALL INTERRUPT ENABLES ON  
\*\*\*\*\*

3797 014246 000004  
3798 014250 012705 014262  
3799 014254 004737 021376  
3800  
3801 014260 000407

TST172: SCOPE  
MOV #ABSF12,R5 ; PTR TO TEST DATA SET  
JSR PC,#ABSF7 ; GO TEST  
  
BR TST173 ;;

3802  
3803 014262  
3804 014262 100000 000001  
3805 014266 000000 000000  
3806 014272 047553 147544  
3807 014276 100014

ABSF12: ; TEST DATA SET ABSF-12:  
.WORD MO,1 ; INITIAL MEM FLOAT NUMBER  
.WORD 0,0 ; EXPECTED FLOAT RESULT  
.WORD 047553,147544 ; FPS: BEFORE, AFTER  
.WORD 100014 ; FEC AFTER ( 0 = N/A )

```

3808
3809
3810
3811
3812
3813
3814 014300 000004
3815 014302 012705 014314
3816 014306 004737 021376
3817
3818 014312 000407
3819
3820 014314
3821 014314 100177 177777
3822 014320 000000 000000
3823 014324 043413 043404
3824 014330 000000
3825
3826
3827

```

```

*****
: TEST 173 TEST OF ABSF INSTR, DATA SET ABSF-13
: * ROUNDING MODE, -0 INTERRUPT ENABLE OFF, ALL OTHERS ON
: *****
TST173: SCOPE
MOV #ABSF13,R5 ; PTR TO TEST DATA SET
JSR PC,@ABSF13 ; GO TEST
BR TST174 ;;

ABSF13: ; TEST DATA SET ABSF-13:
.WORD ZX1MN,M1 ; INITIAL MEM FLOAT NUMBER
.WORD 0.0 ; EXPECTED FLOAT RESULT
.WORD 043413,043404 ; FPS: BEFORE, AFTER
.WORD 000000 ; FEC AFTER ( 0 = N/A )

```

```

3828
3829
3830
3831
3832 014332 000004
3833 014334 012705 014346
3834 014340 004737 021552
3835
3836 014344 000413
3837
3838 014346
3839 014346 000000 000000 000000
3840 014354 000000
3841 014356 000000 000000 000000
3842 014364 000000
3843 014366 047713 047704
3844 014372 000000
3845
3846
3847
3848
3849
3850
3851 014374 000004
3852 014376 012705 014410
3853 014402 004737 021552
3854
3855 014406 000413
3856
3857 014410
3858 014410 052525 052525 052525
3859 014416 052525
3860 014420 052525 052525 052525
3861 014426 052525
3862 014430 047657 047640
3863 014434 000000
3864
3865
3866
3867
3868
3869
3870 014436 000004
3871 014440 012705 014452
3872 014444 004737 021552
3873
3874 014450 000413
3875
3876 014452
3877 014452 125252 125252 125252
3878 014460 125252
3879 014462 025252 125252 125252
3880 014470 125252
3881 014472 047757 047740
3882 014476 000000
3883

```

```

*****
*TEST 174 TEST OF ABSD INSTR, DATA SET ABSD-1
*
* ROUNDING MODE, ALL INTERRUPT ENABLES ON
*****
TST174: SCOPE
MOV #ABSD1,R5 ; PTR TO TEST DATA SET
JSR PC,@ABSDT ; GO TEST
BR TST175 ;;
ABSD1: ; TEST DATA SET ABSD-1:
.WORD 0,0,0,0 ; INITIAL MEM FLOAT NUMBER
.WORD 0,0,0,0 ; EXPECTED FLOAT RESULT
.WORD 047713,047704 ; FPS: BEFORE, AFTER
.WORD 000000 ; FEC AFTER ( 0 = N/A )

*****
*TEST 175 TEST OF ABSD INSTR, DATA SET ABSD-2
*
* TRUNCATE MODE, ALL INTERRUPT ENABLES ON
*****
TST175: SCOPE
MOV #ABSD2,R5 ; PTR TO TEST DATA SET
JSR PC,@ABSDT ; GO TEST
BR TST176 ;;
ABSD2: ; TEST DATA SET ABSD-2:
.WORD ALTP,ALTP,ALTP,ALTP ; INITIAL MEM FLOAT NUMBER
.WORD ALTP,ALTP,ALTP,ALTP ; EXPECTED FLOAT RESULT
.WORD 047657,047640 ; FPS: BEFORE, AFTER
.WORD 000000 ; FEC AFTER ( 0 = N/A )

*****
*TEST 176 TEST OF ABSD INSTR, DATA SET ABSD-3
*
* TRUNCATE MODE, ALL INTERRUPT ENABLES ON
*****
TST176: SCOPE
MOV #ABSD3,R5 ; PTR TO TEST DATA SET
JSR PC,@ABSDT ; GO TEST
BR TST177 ;;
ABSD3: ; TEST DATA SET ABSD-3:
.WORD ALTN,ALTN,ALTN,ALTN ; INITIAL MEM FLOAT NUMBER
.WORD 025252,ALTN,ALTN,ALTN ; EXPECTED FLOAT RESULT
.WORD 047757,047740 ; FPS: BEFORE, AFTER
.WORD 000000 ; FEC AFTER ( 0 = N/A )

```



```

3884
3885
3886
3887
3888
3889 014500 000004
3890 014502 012705 014514
3891 014506 004737 021552
3892
3893 014512 000413
3894
3895 014514
3896 014514 177777 177777 177777
3897 014522 177777
3898 014524 077777 177777 177777
3899 014532 177777
3900 014534 047617 047600
3901 014540 000000
3902
3903
3904
3905
3906
3907
3908 014542 000004
3909 014544 012705 014556
3910 014550 004737 021552
3911
3912 014554 000413
3913
3914 014556
3915 014556 000200 000000 000000
3916 014564 000000
3917 014566 000200 000000 000000
3918 014574 000000
3919 014576 047717 047700
3920 014602 000000
3921
3922
3923
3924
3925
3926
3927 014604 000004
3928 014606 012705 014620
3929 014612 004737 021552
3930
3931 014616 000413
3932
3933 014620
3934 014620 077777 177777 177777
3935 014626 177777
3936 014630 077777 177777 177777
3937 014636 177777
3938 014640 047657 047640
3939 014644 000000

```

```

*****
*TEST 177 TEST OF ABSD INSTR, DATA SET ABSD-4
* ROUNDING MODE, ALL INTERRUPT ENABLES ON
*****
TST177: SCOPE
MOV #ABSD4,R5 ; PTR TO TEST DATA SET
JSR PC,@ABSDT ; GO TEST
BR TST200 ;;
ABSD4: ; TEST DATA SET ABSD-4:
.WORD LGN,M1,M1,M1 ; INITIAL MEM FLOAT NUMBER
.WORD LGP,M1,M1,M1 ; EXPECTED FLOAT RESULT
.WORD 047617,047600 ; FPS: BEFORE, AFTER
.WORD 000000 ; FEC AFTER ( 0 = N/A )
*****
*TEST 200 TEST OF ABSD INSTR, DATA SET ABSD-5
* ROUNDING MODE, ALL INTERRUPT ENABLES ON
*****
TST200: SCOPE
MOV #ABSD5,R5 ; PTR TO TEST DATA SET
JSR PC,@ABSDT ; GO TEST
BR TST201 ;;
ABSD5: ; TEST DATA SET ABSD-5:
.WORD SMP,0,0,0 ; INITIAL MEM FLOAT NUMBER
.WORD SMP,0,0,0 ; EXPECTED FLOAT RESULT
.WORD 047717,047700 ; FPS: BEFORE, AFTER
.WORD 000000 ; FEC AFTER ( 0 = N/A )
*****
*TEST 201 TEST OF ABSD INSTR, DATA SET ABSD-6
* TRUNCATE MODE, ALL INTERRUPT ENABLES ON
*****
TST201: SCOPE
MOV #ABSD6,R5 ; PTR TO TEST DATA SET
JSR PC,@ABSDT ; GO TEST
BR TST202 ;;
ABSD6: ; TEST DATA SET ABSD-6:
.WORD LGP,M1,M1,M1 ; INITIAL MEM FLOAT NUMBER
.WORD LGP,M1,M1,M1 ; EXPECTED FLOAT RESULT
.WORD 047657,047640 ; FPS: BEFORE, AFTER
.WORD 000000 ; FEC AFTER ( 0 = N/A )

```

```

3940
3941
3942
3943
3944
3945
3946 014646 000004
3947 014650 012705 014662
3948 014654 004737 021552
3949
3950 014660 000413
3951
3952 014662
3953 014662 100200 000000 000000
3954 014670 000000
3955 014672 000200 000000 000000
3956 014700 000000
3957 014702 047757 047740
3958 014706 000000
3959
3960
3961
3962
3963
3964
3965 014710 000004
3966 014712 012705 014724
3967 014716 004737 021552
3968
3969 014722 000413
3970
3971 014724
3972 014724 000177 177777 177777
3973 014732 177777
3974 014734 000000 000000 000000
3975 014742 000000
3976 014744 047613 047604
3977 014750 000000
3978
3979
3980
3981
3982
3983
3984 014752 000004
3985 014754 012705 014766
3986 014760 004737 021552
3987
3988 014764 000413
3989
3990 014766
3991 014766 100000 000000 000000
3992 014774 000000
3993 014776 000000 000000 000000
3994 015004 000000
3995 015006 047713 147704

```

```

*****
*TEST 202 TEST OF ABSD INSTR, DATA SET ABSD-7
* TRUNCATE MODE, ALL INTERRUPT ENABLES ON
*****
TST202: SCOPE
MOV #ABSD7,R5 ; PTR TO TEST DATA SET
JSR PC,@ABSDT ; GO TEST
BR TST203 ;;
ABSD7: ; TEST DATA SET ABSD-7:
.WORD SMN,0,0,0 ; INITIAL MEM FLOAT NUMBER
.WORD SMP,0,0,0 ; EXPECTED FLOAT RESULT
.WORD 047757,047740 ; FPS: BEFORE, AFTER
.WORD 000000 ; FEC AFTER ( 0 = N/A )

*****
*TEST 203 TEST OF ABSD INSTR, DATA SET ABSD-10
* ROUNDING MODE, ALL INTERRUPT ENABLES ON
*****
TST203: SCOPE
MOV #ABSD10,R5 ; PTR TO TEST DATA SET
JSR PC,@ABSDT ; GO TEST
BR TST204 ;;
ABSD10: ; TEST DATA SET ABSD-10:
.WORD ZXIMP,M1,M1,M1 ; INITIAL MEM FLOAT NUMBER
.WORD 0,0,0,0 ; EXPECTED FLOAT RESULT
.WORD 047613,047604 ; FPS: BEFORE, AFTER
.WORD 000000 ; FEC AFTER ( 0 = N/A )

*****
*TEST 204 TEST OF ABSD INSTR, DATA SET ABSD-11
* ROUNDING MODE, ALL INTERRUPT ENABLES ON
*****
TST204: SCOPE
MOV #ABSD11,R5 ; PTR TO TEST DATA SET
JSR PC,@ABSDT ; GO TEST
BR TST205 ;;
ABSD11: ; TEST DATA SET ABSD-11:
.WORD M0,0,0,0 ; INITIAL MEM FLOAT NUMBER
.WORD 0,0,0,0 ; EXPECTED FLOAT RESULT
.WORD 047713,147704 ; FPS: BEFORE, AFTER

```

E07

FPU BASIC INSTR TESTS MACY11 27(1006) 09-FEB-77 09:48 PAGE 84  
DGFPA.P11 09-FEB-77 09:42 T204 TEST OF ABSD INSTR, DATA SET ABSD-11

3996 015012 100014 .WORD 100014 ; FEC AFTER ( 0 = N/A )

3997  
3998  
3999  
4000  
4001  
4002  
4003  
4004  
4005  
4006  
4007  
4008  
4009  
4010  
4011  
4012  
4013  
4014  
4015  
4016  
4017  
4018  
4019  
4020  
4021  
4022  
4023  
4024  
4025  
4026  
4027  
4028  
4029  
4030  
4031  
4032  
4033  
4034  
4035  
4036  
4037

015014 000004  
015016 012705 015030  
015022 004737 021552

\*\*\*\*\*  
: TEST 205 TEST OF ABSD INSTR, DATA SET ABSD-12  
: \* TRUNCATE MODE, ALL INTERRUPT ENABLES ON  
: \*\*\*\*\*

TST205: SCOPE  
MOV #ABSD12,R5 ; PTR TO TEST DATA SET  
JSR PC,#ABSDT ; GO TEST  
BR TST206 ;;

ABSD12: ; TEST DATA SET ABSD-12:  
.WORD MO,0,0,1 ; INITIAL MEM FLOAT NUMBER  
.WORD 0,0,0,0 ; EXPECTED FLOAT RESULT  
.WORD 047653,147644 ; FPS: BEFORE, AFTER  
.WORD 100014 ; FEC AFTER ( 0 = N/A )

\*\*\*\*\*  
: TEST 206 TEST OF ABSD INSTR, DATA SET ABSD-13  
: \* TRUNCATE MODE, -0 INTERRUPT ENABLE OFF, ALL OTHERS ON  
: \*\*\*\*\*

TST206: SCOPE  
MOV #ABSD13,R5 ; PTR TO TEST DATA SET  
JSR PC,#ABSDT ; GO TEST  
BR TST207 ;;

ABSD13: ; TEST DATA SET ABSD-13:  
.WORD ZX1MN,M1,M1,M1 ; INITIAL MEM FLOAT NUMBER  
.WORD 0,0,0,0 ; EXPECTED FLOAT RESULT  
.WORD 043653,043644 ; FPS: BEFORE, AFTER  
.WORD 000000 ; FEC AFTER ( 0 = N/A )

F07

FPU BASIC INSTR TESTS MACY11 27(1006) 09-FEB-77 09:48 PAGE 85  
DQFPAA.P11 09-FEB-77 09:42

T207 TEST OF NEGF INSTR, DATA SET NEGF-1

4038  
4039  
4040  
4041  
4042 015120 000004  
4043 015122 012705 015134  
4044 015126 004737 021762  
4045  
4046 015132 000407  
4047  
4048 015134  
4049 015134 000000 000000  
4050 015140 000000 000000  
4051 015144 047413 047404  
4052 015150 000000  
4053  
4054  
4055  
4056  
4057  
4058  
4059 015152 000004  
4060 015154 012705 015166  
4061 015160 004737 021762  
4062  
4063 015164 000407  
4064  
4065 015166  
4066 015166 052525 052525  
4067 015172 152525 052525  
4068 015176 047547 047550  
4069 015202 000000  
4070  
4071  
4072  
4073  
4074  
4075  
4076 015204 000004  
4077 015206 012705 015220  
4078 015212 004737 021762  
4079  
4080 015216 000407  
4081  
4082 015220  
4083 015220 125252 125252  
4084 015224 025252 125252  
4085 015230 047517 047500  
4086 015234 000000  
4087  
4088  
4089  
4090  
4091  
4092  
4093 015236 000004

\*\*\*\*\*  
\*TEST 207 TEST OF NEGF INSTR, DATA SET NEGF-1  
\* ROUNDDING MODE, ALL INTERRUPT ENABLES ON  
\*\*\*\*\*

TST207: SCOPE  
MOV #NEGF1,R5 ; PTR TO TEST DATA SET  
JSR PC,@#NEGFT ; GO TEST  
BR TST210 ;;

NEGF1: ; TEST DATA SET NEGF-1:  
.WORD 0,0 ; INITIAL MEM FLOAT NUMBER  
.WORD 0,0 ; EXPECTED FLOAT RESULT  
.WORD 047413,047404 ; FPS: BEFORE, AFTER  
.WORD 000000 ; FEC AFTER ( 0 = N/A )

\*\*\*\*\*  
\*TEST 210 TEST OF NEGF INSTR, DATA SET NEGF-2  
\* TRUNCATE MODE, ALL INTERRUPT ENABLES ON  
\*\*\*\*\*

TST210: SCOPE  
MOV #NEGF2,R5 ; PTR TO TEST DATA SET  
JSR PC,@#NEGFT ; GO TEST  
BR TST211 ;;

NEGF2: ; TEST DATA SET NEGF-2:  
.WORD ALTP,ALTP ; INITIAL MEM FLOAT NUMBER  
.WORD 152525,ALTP ; EXPECTED FLOAT RESULT  
.WORD 047547,047550 ; FPS: BEFORE, AFTER  
.WORD 000000 ; FEC AFTER ( 0 = N/A )

\*\*\*\*\*  
\*TEST 211 TEST OF NEGF INSTR, DATA SET NEGF-3  
\* ROUNDDING MODE, ALL INTERRUPT ENABLES ON  
\*\*\*\*\*

TST211: SCOPE  
MOV #NEGF3,R5 ; PTR TO TEST DATA SET  
JSR PC,@#NEGFT ; GO TEST  
BR TST212 ;;

NEGF3: ; TEST DATA SET NEGF-3:  
.WORD ALTN,ALTN ; INITIAL MEM FLOAT NUMBER  
.WORD 025252,ALTN ; EXPECTED FLOAT RESULT  
.WORD 047517,047500 ; FPS: BEFORE, AFTER  
.WORD 000000 ; FEC AFTER ( 0 = N/A )

\*\*\*\*\*  
\*TEST 212 TEST OF NEGF INSTR, DATA SET NEGF-4  
\* TRUNCATE MODE, ALL INTERRUPT ENABLES ON  
\*\*\*\*\*

TST212: SCOPE

G07

FPU BASIC INSTR TESTS MACY11 27(1006) 09-FEB-77 09:48 PAGE 86  
DGFPA.P11 09-FEB-77 09:42 T212 TEST OF NEG F INSTR, DATA SET NEG F-4

4094 015240 012705 015252  
4095 015244 004737 021762  
4096  
4097 015250 000407  
4098  
4099 015252  
4100 015252 177777 177777  
4101 015256 077777 177777  
4102 015262 047457 047440  
4103 015266 000000  
4104  
4105  
4106  
4107  
4108  
4109  
4110 015270 000004  
4111 015272 012705 015304  
4112 015276 004737 021762  
4113  
4114 015302 000407  
4115  
4116 015304  
4117 015304 000200 000000  
4118 015310 100200 000000  
4119 015314 047407 047410  
4120 015320 000000  
4121  
4122  
4123  
4124  
4125  
4126  
4127 015322 000004  
4128 015324 012705 015336  
4129 015330 004737 021762  
4130  
4131 015334 000407  
4132  
4133 015336  
4134 015336 077777 177777  
4135 015342 177777 177777  
4136 015346 047547 047550  
4137 015352 000000  
4138  
4139  
4140  
4141  
4142  
4143  
4144 015354 000004  
4145 015356 012705 015370  
4146 015362 004737 021762  
4147  
4148 015366 000407  
4149

MOV #NEGF4,R5 ; PTR TO TEST DATA SET  
JSR PC,@#NEGFT ; GO TEST  
BR TST213 ;;

NEGF4: ; TEST DATA SET NEG F-4:  
.WORD LGN,M1 ; INITIAL MEM FLOAT NUMBER  
.WORD LGP,M1 ; EXPECTED FLOAT RESULT  
.WORD 047457,047440 ; FPS: BEFORE, AFTER  
.WORD 000000 ; FEC AFTER ( 0 = N/A )

\*\*\*\*\*  
\*TEST 213 TEST OF NEG F INSTR, DATA SET NEG F-5  
\* ROUNDDING MODE, ALL INTERRUPT ENABLES ON  
\*\*\*\*\*

TST213: SCOPE  
MOV #NEGF5,R5 ; PTR TO TEST DATA SET  
JSR PC,@#NEGFT ; GO TEST  
BR TST214 ;;

NEGF5: ; TEST DATA SET NEG F-5:  
.WORD SMP,0 ; INITIAL MEM FLOAT NUMBER  
.WORD SMN,0 ; EXPECTED FLOAT RESULT  
.WORD 047407,047410 ; FPS: BEFORE, AFTER  
.WORD 000000 ; FEC AFTER ( 0 = N/A )

\*\*\*\*\*  
\*TEST 214 TEST OF NEG F INSTR, DATA SET NEG F-6  
\* TRUNCATE MODE, ALL INTERRUPT ENABLES ON  
\*\*\*\*\*

TST214: SCOPE  
MOV #NEGF6,R5 ; PTR TO TEST DATA SET  
JSR PC,@#NEGFT ; GO TEST  
BR TST215 ;;

NEGF6: ; TEST DATA SET NEG F-6:  
.WORD LGP,M1 ; INITIAL MEM FLOAT NUMBER  
.WORD LGN,M1 ; EXPECTED FLOAT RESULT  
.WORD 047547,047550 ; FPS: BEFORE, AFTER  
.WORD 000000 ; FEC AFTER ( 0 = N/A )

\*\*\*\*\*  
\*TEST 215 TEST OF NEG F INSTR, DATA SET NEG F-7  
\* ROUNDDING MODE, ALL INTERRUPT ENABLES ON  
\*\*\*\*\*

TST215: SCOPE  
MOV #NEGF7,R5 ; PTR TO TEST DATA SET  
JSR PC,@#NEGFT ; GO TEST  
BR TST216 ;;

4150	015370		
4151	015370	100200	000000
4152	015374	000200	000000
4153	015400	047517	047500
4154	015404	000000	

```

NEGF7: ; TEST DATA SET NEGF-7:
.WORD SMN,0 ; INITIAL MEM FLOAT NUMBER
.WORD SMP,0 ; EXPECTED FLOAT RESULT
.WORD 047517,047500 ; FPS: BEFORE, AFTER
.WORD 000000 ; FEC AFTER ( 0 = N/A )

```

4155			
4156			
4157			
4158			
4159			
4160			

```

*****
; *TEST 216 TEST OF NEGF INSTR, DATA SET NEGF-10
; * TRUNCATE MODE, ALL INTERRUPT ENABLES ON
*****

```

4161	015406	000004	
4162	015410	012705	015422
4163	015414	004737	021762
4164			
4165	015420	000407	
4166			

```

TST216: SCOPE
MOV #NEGF10,R5 ; PTR TO TEST DATA SET
JSR PC,@#NEGF1 ; GO TEST
BR TST217 ;;

```

4167	015422		
4168	015422	000177	177777
4169	015426	000000	000000
4170	015432	047453	047444
4171	015436	000000	

```

NEGF10: ; TEST DATA SET NEGF-10:
.WORD ZXIMP,M1 ; INITIAL MEM FLOAT NUMBER
.WORD 0,0 ; EXPECTED FLOAT RESULT
.WORD 047453,047444 ; FPS: BEFORE, AFTER
.WORD 000000 ; FEC AFTER ( 0 = N/A )

```

4172			
4173			
4174			
4175			
4176			
4177			

```

*****
; *TEST 217 TEST OF NEGF INSTR, DATA SET NEGF-11
; * TRUNCATE MODE, ALL INTERRUPT ENABLES ON
*****

```

4178	015440	000004	
4179	015442	012705	015454
4180	015446	004737	021762
4181			
4182	015452	000407	
4183			

```

TST217: SCOPE
MOV #NEGF11,R5 ; PTR TO TEST DATA SET
JSR PC,@#NEGF1 ; GO TEST
BR TST220 ;;

```

4184	015454		
4185	015454	100000	000000
4186	015460	000000	000000
4187	015464	047453	147444
4188	015470	100014	

```

NEGF11: ; TEST DATA SET NEGF-11:
.WORD MO,0 ; INITIAL MEM FLOAT NUMBER
.WORD 0,0 ; EXPECTED FLOAT RESULT
.WORD 047453,147444 ; FPS: BEFORE, AFTER
.WORD 100014 ; FEC AFTER ( 0 = N/A )

```

4189			
4190			
4191			
4192			
4193			
4194			

```

*****
; *TEST 220 TEST OF NEGF INSTR, DATA SET NEGF-12
; * TRUNCATE MODE, ALL INTERRUPT ENABLES ON
*****

```

4195	015472	000004	
4196	015474	012705	015506
4197	015500	004737	021762
4198			
4199	015504	000407	

```

TST220: SCOPE
MOV #NEGF12,R5 ; PTR TO TEST DATA SET
JSR PC,@#NEGF1 ; GO TEST
BR TST221 ;;

```

4200			
4201	015506		
4202	015506	100000	000001
4203	015512	000000	000000
4204	015516	047553	147544
4205	015522	100014	

```

NEGF12: ; TEST DATA SET NEGF-12:
.WORD MO,1 ; INITIAL MEM FLOAT NUMBER
.WORD 0,0 ; EXPECTED FLOAT RESULT
.WORD 047553,147544 ; FPS: BEFORE, AFTER
.WORD 100014 ; FEC AFTER ( 0 = N/A )

```

```

4206
4207
4208
4209
4210
4211
4212 015524 000004
4213 015526 012705 015540
4214 015532 004737 021762
4215
4216 015536 000407
4217
4218 015540
4219 015540 100177 177777
4220 015544 000000 000000
4221 015550 043513 043504
4222 015554 000000
4223
4224
4225

```

```

;*****
;*TEST 221 TEST OF NEGF INSTR, DATA SET NEGF-13
;* ROUNDING MODE, -0 INTERRUPT ENABLE OFF, ALL OTHERS ON
;*****
TST221: SCOPE
MOV #NEGF13,R5 ; PTR TO TEST DATA SET
JSR PC,@#NEGF1 ; GO TEST
BR TST222 ;;

NEGF13: ; TEST DATA SET NEGF-13:
.WORD ZX1MN,M1 ; INITIAL MEM FLOAT NUMBER
.WORD 0.0 ; EXPECTED FLOAT RESULT
.WORD 043513,043504 ; FPS: BEFORE, AFTER
.WORD 000000 ; FEC AFTER ( 0 = N/A )

```

```

4226
4227
4228
4229
4230 015556 000004
4231 015560 012705 015572
4232 015564 004737 022136
4233
4234 015570 000413
4235
4236 015572
4237 015572 000000 000000 000000
4238 015600 000000
4239 015602 000000 000000 000000
4240 015610 000000
4241 015612 047753 047744
4242 015616 000000
4243
4244
4245
4246
4247
4248
4249 015620 000004
4250 015622 012705 015634
4251 015626 004737 022136
4252
4253 015632 000413
4254
4255 015634
4256 015634 052525 052525 052525
4257 015642 052525
4258 015644 152525 052525 052525
4259 015652 052525
4260 015654 047607 047610
4261 015660 000000
4262
4263
4264
4265
4266
4267
4268 015662 000004
4269 015664 012705 015676
4270 015670 004737 022136
4271
4272 015674 000413
4273
4274 015676
4275 015676 125252 125252 125252
4276 015704 125252
4277 015706 025252 125252 125252
4278 015714 125252
4279 015716 047657 047640
4280 015722 000000
4281

```

```

*****
; TEST 222 TEST OF NEGD INSTR, DATA SET NEGD-1
;* TRUNCATE MODE, ALL INTERRUPT ENABLES ON
*****
TST222: SCOPE
MOV #NEGD1,R5 ; PTR TO TEST DATA SET
JSR PC,@#NEGDT ; GO TEST
BR TST223 ;;

NEGD1: ; TEST DATA SET NEGD-1:
.WORD 0,0,0,0 ; INITIAL MEM FLOAT NUMBER
.WORD 0,0,0,0 ; EXPECTED FLOAT RESULT
.WORD 047753,047744 ; FPS: BEFORE, AFTER
.WORD 000000 ; FEC AFTER ( 0 = N/A )

*****
; TEST 223 TEST OF NEGD INSTR, DATA SET NEGD-2
;* ROUNDING MODE, ALL INTERRUPT ENABLES ON
*****
TST223: SCOPE
MOV #NEGD2,R5 ; PTR TO TEST DATA SET
JSR PC,@#NEGDT ; GO TEST
BR TST224 ;;

NEGD2: ; TEST DATA SET NEGD-2:
.WORD ALTP,ALTP,ALTP,ALTP ; INITIAL MEM FLOAT NUMBER
.WORD 152525,ALTP,ALTP,ALTP ; EXPECTED FLOAT RESULT
.WORD 047607,047610 ; FPS: BEFORE, AFTER
.WORD 000000 ; FEC AFTER ( 0 = N/A )

*****
; TEST 224 TEST OF NEGD INSTR, DATA SET NEGD-3
;* TRUNCATE MODE, ALL INTERRUPT ENABLES ON
*****
TST224: SCOPE
MOV #NEGD3,R5 ; PTR TO TEST DATA SET
JSR PC,@#NEGDT ; GO TEST
BR TST225 ;;

NEGD3: ; TEST DATA SET NEGD-3:
.WORD ALTN,ALTN,ALTN,ALTN ; INITIAL MEM FLOAT NUMBER
.WORD 025252,ALTN,ALTN,ALTN ; EXPECTED FLOAT RESULT
.WORD 047657,047640 ; FPS: BEFORE, AFTER
.WORD 000000 ; FEC AFTER ( 0 = N/A )

```



```

4282
4283
4284
4285
4286
4287 015724 000004
4288 015726 012705 015740
4289 015732 004737 022136
4290
4291 015736 000413
4292
4293 015740
4294 015740 177777 177777 177777
4295 015746 177777
4296 015750 077777 177777 177777
4297 015756 177777
4298 015760 047717 047700
4299 015764 000000
4300
4301
4302
4303
4304
4305
4306 015766 000004
4307 015770 012705 016002
4308 015774 004737 022136
4309
4310 016000 000413
4311
4312 016002
4313 016002 000200 000000 000000
4314 016010 000000
4315 016012 100200 000000 000000
4316 016020 000000
4317 016022 047747 047750
4318 016026 000000
4319
4320
4321
4322
4323
4324
4325 016030 000004
4326 016032 012705 016044
4327 016036 004737 022136
4328
4329 016042 000413
4330
4331 016044
4332 016044 077777 177777 177777
4333 016052 177777
4334 016054 177777 177777 177777
4335 016062 177777
4336 016064 047607 047610
4337 016070 000000

```

```

*****
*TEST 225 TEST OF NEGD INSTR, DATA SET NEGD-4
* ROUNDING MODE, ALL INTERRUPT ENABLES ON
*****
TST225: SCOPE
MOV #NEGD4,R5 ; PTR TO TEST DATA SET
JSR PC,@#NEGDT ; GO TEST
BR TST226 ;;
NEGD4: ; TEST DATA SET NEGD-4:
.WORD LGN,M1,M1,M1 ; INITIAL MEM FLOAT NUMBER
.WORD LGP,M1,M1,M1 ; EXPECTED FLOAT RESULT
.WORD 047717,047700 ; FPS: BEFORE, AFTER
.WORD 000000 ; FEC AFTER ( 0 = N/A )
*****
*TEST 226 TEST OF NEGD INSTR, DATA SET NEGD-5
* TRUNCATE MODE, ALL INTERRUPT ENABLES ON
*****
TST226: SCOPE
MOV #NEGD5,R5 ; PTR TO TEST DATA SET
JSR PC,@#NEGDT ; GO TEST
BR TST227 ;;
NEGD5: ; TEST DATA SET NEGD-5:
.WORD SMP,0,0,0 ; INITIAL MEM FLOAT NUMBER
.WORD SMN,0,0,0 ; EXPECTED FLOAT RESULT
.WORD 047747,047750 ; FPS: BEFORE, AFTER
.WORD 000000 ; FEC AFTER ( 0 = N/A )
*****
*TEST 227 TEST OF NEGD INSTR, DATA SET NEGD-6
* ROUNDING MODE, ALL INTERRUPT ENABLES ON
*****
TST227: SCOPE
MOV #NEGD6,R5 ; PTR TO TEST DATA SET
JSR PC,@#NEGDT ; GO TEST
BR TST230 ;;
NEGD6: ; TEST DATA SET NEGD-6:
.WORD LGP,M1,M1,M1 ; INITIAL MEM FLOAT NUMBER
.WORD LGN,M1,M1,M1 ; EXPECTED FLOAT RESULT
.WORD 047607,047610 ; FPS: BEFORE, AFTER
.WORD 000000 ; FEC AFTER ( 0 = N/A )

```

```

4338
4339
4340
4341
4342
4343
4344 016072 000004
4345 016074 012705 016106
4346 016100 004737 022136
4347
4348 016104 000413
4349
4350 016106
4351 016106 100200 000000 000000
4352 016114 000000
4353 016116 000200 000000 000000
4354 016124 000000
4355 016126 047657 047640
4356 016132 000000
4357
4358
4359
4360
4361
4362
4363 016134 000004
4364 016136 012705 016150
4365 016142 004737 022136
4366
4367 016146 000413
4368
4369 016150
4370 016150 000177 177777 177777
4371 016156 177777
4372 016160 000000 000000 000000
4373 016166 000000
4374 016170 047713 047704
4375 016174 000000
4376
4377
4378
4379
4380
4381
4382 016176 000004
4383 016200 012705 016212
4384 016204 004737 022136
4385
4386 016210 000413
4387
4388 016212
4389 016212 100000 000000 000000
4390 016220 000000
4391 016222 000000 000000 000000
4392 016230 000000
4393 016232 047713 147704

```

```

*****
*TEST 230 TEST OF NEG D INSTR, DATA SET NEG D-7
* TRUNCATE MODE, ALL INTERRUPT ENABLES ON
*****
TST230: SCOPE
MOV #NEG D7,R5 ; PTR TO TEST DATA SET
JSR PC,@#NEG D7 ; GO TEST
BR TST231 ;;
NEG D7: ; TEST DATA SET NEG D-7:
.WORD SMN,0,0,0 ; INITIAL MEM FLOAT NUMBER
.WORD SMP,0,0,0 ; EXPECTED FLOAT RESULT
.WORD 047657,047640 ; FPS: BEFORE, AFTER
.WORD 000000 ; FEC AFTER ( 0 = N/A )
*****
*TEST 231 TEST OF NEG D INSTR, DATA SET NEG D-10
* ROUNDING MODE, ALL INTERRUPT ENABLES ON
*****
TST231: SCOPE
MOV #NEG D10,R5 ; PTR TO TEST DATA SET
JSR PC,@#NEG D10 ; GO TEST
BR TST232 ;;
NEG D10: ; TEST DATA SET NEG D-10:
.WORD ZX1MP,M1,M1,M1 ; INITIAL MEM FLOAT NUMBER
.WORD 0,0,0,0 ; EXPECTED FLOAT RESULT
.WORD 047713,047704 ; FPS: BEFORE, AFTER
.WORD 000000 ; FEC AFTER ( 0 = N/A )
*****
*TEST 232 TEST OF NEG D INSTR, DATA SET NEG D-11
* ROUNDING MODE, ALL INTERRUPT ENABLES ON
*****
TST232: SCOPE
MOV #NEG D11,R5 ; PTR TO TEST DATA SET
JSR PC,@#NEG D11 ; GO TEST
BR TST233 ;;
NEG D11: ; TEST DATA SET NEG D-11:
.WORD MD,0,0,0 ; INITIAL MEM FLOAT NUMBER
.WORD 0,0,0,0 ; EXPECTED FLOAT RESULT
.WORD 047713,147704 ; FPS: BEFORE, AFTER

```

M07

FPU BASIC INSTR TESTS MACY11 27(1006) 09-FEB-77 09:48 PAGE 92  
DQFPA.P11 09-FEB-77 09:42 T232 TEST OF NEG D INSTR, DATA SET NEG D-11

4394 016236 100014 .WORD 100014 ; FEC AFTER ( 0 = N/A )

4395  
4396  
4397  
4398  
4399  
4400  
\*\*\*\*\*  
\*TEST 233 TEST OF NEG D INSTR, DATA SET NEG D-12  
\* ROUNDING MODE, ALL INTERRUPT ENABLES ON  
\*\*\*\*\*

4401 016240 000004  
4402 016242 012705 016254  
4403 016246 004737 022136  
4404  
4405 016252 000413  
4406  
4407  
4408  
4409  
4410  
4411  
4412  
4413  
4414  
4415  
4416  
4417  
4418  
4419

TST233: SCOPE  
MOV #NEG D12,R5 ; PTR TO TEST DATA SET  
JSR PC,#NEG D T ; GO TEST  
BR TST234 ; ;  
NEG D12: ; TEST DATA SET NEG D-12:  
.WORD MO,0,0,1 ; INITIAL MEM FLOAT NUMBER  
.WORD 0,0,0,0 ; EXPECTED FLOAT RESULT  
.WORD 047613,147604 ; FPS: BEFORE, AFTER  
.WORD 100014 ; FEC AFTER ( 0 = N/A )

4420  
4421  
4422  
4423  
4424  
4425  
4426  
4427  
4428  
4429  
4430  
4431  
4432  
4433  
4434  
4435  
\*\*\*\*\*  
\*TEST 234 TEST OF NEG D INSTR, DATA SET NEG D-13  
\* TRUNCATE MODE, -0 INTERRUPT ENABLE OFF, ALL OTHERS ON  
\*\*\*\*\*

4420 016302 000004  
4421 016304 012705 016316  
4422 016310 004737 022136  
4423  
4424 016314 000413  
4425  
4426 016316  
4427 016316 100177 177777 177777  
4428 016324 177777  
4429 016326 000000 000000 000000  
4430 016334 000000  
4431 016336 043753 043744  
4432 016342 000000  
4433  
4434  
4435

TST234: SCOPE  
MOV #NEG D13,R5 ; PTR TO TEST DATA SET  
JSR PC,#NEG D T ; GO TEST  
BR TST235 ; ;  
NEG D13: ; TEST DATA SET NEG D-13:  
.WORD ZX1MN,M1,M1,M1 ; INITIAL MEM FLOAT NUMBER  
.WORD 0,0,0,0 ; EXPECTED FLOAT RESULT  
.WORD 043753,043744 ; FPS: BEFORE, AFTER  
.WORD 000000 ; FEC AFTER ( 0 = N/A )

```

4436
4437
4438
4439
4440 016344 000004
4441 016346 012705 016360
4442 016352 004737 022346
4443
4444 016356 000407
4445
4446 016360
4447 016360 000000 000000
4448 016364 000000 000000
4449 016370 047553 047544
4450 016374 000000
4451
4452
4453
4454
4455
4456
4457 016376 000004
4458 016400 012705 016412
4459 016404 004737 022346
4460
4461 016410 000407
4462
4463 016412
4464 016412 177777 177777
4465 016416 000000 000000
4466 016422 047413 047404
4467 016426 000000
4468
4469
4470
4471
4472
4473
4474 016430 000004
4475 016432 012705 016444
4476 016436 004737 022346
4477
4478 016442 000407
4479
4480 016444
4481 016444 052525 052525
4482 016450 000000 000000
4483 016454 047453 047444
4484 016460 000000
4485
4486
4487
4488
4489
4490
4491 016462 000004

```

```

*****
*TEST 235 TEST OF CLRF INSTR, DATA SET CLRF-1
* TRUNCATE MODE, ALL INTERRUPT ENABLES ON
*****

```

```

†ST235: SCOPE
MOV #CLRF1,R5 ; PTR TO TEST DATA SET
JSR PC,@#CLRFT ; GO TEST
BR TST236 ;;

```

```

CLRF1: ; TEST DATA SET CLRF-1:
.WORD 0,0 ; INITIAL MEM FLOAT NUMBER
.WORD 0,0 ; EXPECTED FLOAT RESULT
.WORD 047553,047544 ; FPS: BEFORE, AFTER
.WORD 000000 ; FEC AFTER ( 0 = N/A )

```

```

*****
*TEST 236 TEST OF CLRF INSTR, DATA SET CLRF-2
* ROUNDING MODE, ALL INTERRUPT ENABLES ON
*****

```

```

†ST236: SCOPE
MOV #CLRF2,R5 ; PTR TO TEST DATA SET
JSR PC,@#CLRFT ; GO TEST
BR TST237 ;;

```

```

CLRF2: ; TEST DATA SET CLRF-2:
.WORD M1,M1 ; INITIAL MEM FLOAT NUMBER
.WORD 0,0 ; EXPECTED FLOAT RESULT
.WORD 047413,047404 ; FPS: BEFORE, AFTER
.WORD 000000 ; FEC AFTER ( 0 = N/A )

```

```

*****
*TEST 237 TEST OF CLRF INSTR, DATA SET CLRF-3
* TRUNCATE MODE, ALL INTERRUPT ENABLES ON
*****

```

```

†ST237: SCOPE
MOV #CLRF3,R5 ; PTR TO TEST DATA SET
JSR PC,@#CLRFT ; GO TEST
BR TST240 ;;

```

```

CLRF3: ; TEST DATA SET CLRF-3:
.WORD ALTP,ALTP ; INITIAL MEM FLOAT NUMBER
.WORD 0,0 ; EXPECTED FLOAT RESULT
.WORD 047453,047444 ; FPS: BEFORE, AFTER
.WORD 000000 ; FEC AFTER ( 0 = N/A )

```

```

*****
*TEST 240 TEST OF CLRF INSTR, DATA SET CLRF-4
* ROUNDING MODE, ALL INTERRUPT ENABLES ON
*****

```

```

†ST240: SCOPE

```

4492 016464 012705 016476  
4493 016470 004737 022346  
4494  
4495 016474 000407  
4496  
4497 016476  
4498 016476 125252 125252  
4499 016502 000000 000000  
4500 016506 047513 047504  
4501 016512 000000  
4502  
4503  
4504  
4505  
4506  
4507  
4508 016514 000004  
4509 016516 012705 016530  
4510 016522 004737 022346  
4511  
4512 016526 000407  
4513  
4514 016530  
4515 016530 100000 000000  
4516 016534 000000 000000  
4517 016540 047553 047544  
4518 016544 000000  
4519  
4520  
4521  
4522  
4523  
4524  
4525 016546 000004  
4526 016550 012705 016562  
4527 016554 004737 022346  
4528  
4529 016560 000407  
4530  
4531 016562  
4532 016562 000177 177777  
4533 016566 000000 000000  
4534 016572 047413 047404  
4535 016576 000000  
4536  
4537  
4538

MOV #CLRF4,R5 ; PTR TO TEST DATA SET  
JSR PC,@#CLRFT ; GO TEST  
BR TST241 ; ;  
CLRF4: ; TEST DATA SET CLRF-4:  
.WORD ALTN,ALTN ; INITIAL MEM FLOAT NUMBER  
.WORD 0,0 ; EXPECTED FLOAT RESULT  
.WORD 047513,047504 ; FPS: BEFORE, AFTER  
.WORD 000000 ; FEC AFTER ( 0 = N/A )

\*\*\*\*\*  
\*TEST 241 TEST OF CLRF INSTR, DATA SET CLRF-5  
\* TRUNCATE MODE, ALL INTERRUPT ENABLES ON  
\*\*\*\*\*

TST241: SCOPE  
MOV #CLRF5,R5 ; PTR TO TEST DATA SET  
JSR PC,@#CLRFT ; GO TEST  
BR TST242 ; ;

CLRF5: ; TEST DATA SET CLRF-5:  
.WORD 0,0 ; INITIAL MEM FLOAT NUMBER  
.WORD 0,0 ; EXPECTED FLOAT RESULT  
.WORD 047553,047544 ; FPS: BEFORE, AFTER  
.WORD 000000 ; FEC AFTER ( 0 = N/A )

\*\*\*\*\*  
\*TEST 242 TEST OF CLRF INSTR, DATA SET CLRF-6  
\* ROUNDING MODE, ALL INTERRUPT ENABLES ON  
\*\*\*\*\*

TST242: SCOPE  
MOV #CLRF6,R5 ; PTR TO TEST DATA SET  
JSR PC,@#CLRFT ; GO TEST  
BR TST243 ; ;

CLRF6: ; TEST DATA SET CLRF-6:  
.WORD ZXIMP,M1 ; INITIAL MEM FLOAT NUMBER  
.WORD 0,0 ; EXPECTED FLOAT RESULT  
.WORD 047413,047404 ; FPS: BEFORE, AFTER  
.WORD 000000 ; FEC AFTER ( 0 = N/A )

```

4539
4540
4541
4542
4543 016600 000004
4544 016602 012705 016614
4545 016606 004737 022522
4546
4547 016612 000413
4548
4549 016614
4550 016614 000000 000000 000000
4551 016622 000000
4552 016624 000000 000000 000000
4553 016632 000000
4554 016634 047613 047604
4555 016640 000000
4556
4557
4558
4559
4560
4561
4562 016642 000004
4563 016644 012705 016656
4564 016650 004737 022522
4565
4566 016654 000413
4567
4568 016656
4569 016656 177777 177777 177777
4570 016664 177777
4571 016666 000000 000000 000000
4572 016674 000000
4573 016676 047753 047744
4574 016702 000000
4575
4576
4577
4578
4579
4580
4581 016704 000004
4582 016706 012705 016720
4583 016712 004737 022522
4584
4585 016716 000413
4586
4587 016720
4588 016720 052525 052525 052525
4589 016726 052525
4590 016730 000000 000000 000000
4591 016736 000000
4592 016740 047653 047644
4593 016744 000000
4594

```

```

*****
;TEST 243 TEST OF CLRD INSTR, DATA SET CLRD-1
;*
; ROUNDING MODE, ALL INTERRUPT ENABLES ON
*****

```

```

†TST243: SCOPE
MOV #CLRD1,R5 ; PTR TO TEST DATA SET
JSR PC,@#CLRDT ; GO TEST
BR TST244 ;;

```

```

CLRD1: ; TEST DATA SET CLRD-1:
.WORD 0,0,0,0 ; INITIAL MEM FLOAT NUMBER
.WORD 0,0,0,0 ; EXPECTED FLOAT RESULT
.WORD 047613,047604 ; FPS: BEFORE, AFTER
.WORD 000000 ; FEC AFTER ( 0 = N/A )

```

```

*****
;TEST 244 TEST OF CLRD INSTR, DATA SET CLRD-2
;*
; TRUNCATE MODE, ALL INTERRUPT ENABLES ON
*****

```

```

†TST244: SCOPE
MOV #CLRD2,R5 ; PTR TO TEST DATA SET
JSR PC,@#CLRDT ; GO TEST
BR TST245 ;;

```

```

CLRD2: ; TEST DATA SET CLRD-2:
.WORD M1,M1,M1,M1 ; INITIAL MEM FLOAT NUMBER
.WORD 0,0,0,0 ; EXPECTED FLOAT RESULT
.WORD 047753,047744 ; FPS: BEFORE, AFTER
.WORD 000000 ; FEC AFTER ( 0 = N/A )

```

```

*****
;TEST 245 TEST OF CLRD INSTR, DATA SET CLRD-3
;*
; TRUNCATE MODE, ALL INTERRUPT ENABLES ON
*****

```

```

†TST245: SCOPE
MOV #CLRD3,R5 ; PTR TO TEST DATA SET
JSR PC,@#CLRDT ; GO TEST
BR TST246 ;;

```

```

CLRD3: ; TEST DATA SET CLRD-3:
.WORD ALTP,ALTP,ALTP,ALTP ; INITIAL MEM FLOAT NUMBER
.WORD 0,0,0,0 ; EXPECTED FLOAT RESULT
.WORD 047653,047644 ; FPS: BEFORE, AFTER
.WORD 000000 ; FEC AFTER ( 0 = N/A )

```

```

4595
4596
4597
4598
4599
4600 016746 000004
4601 016750 012705 016762
4602 016754 004737 022522
4603
4604 016760 000413
4605
4606 016762
4607 016762 125252 125252 125252
4608 016770 125252
4609 016772 000000 000000 000000
4610 017000 000000
4611 017002 047713 047704
4612 017006 000000
4613
4614
4615
4616
4617
4618
4619 017010 000004
4620 017012 012705 017024
4621 017016 004737 022522
4622
4623 017022 000413
4624
4625 017024
4626 017024 100000 000000 000000
4627 017032 000000
4628 017034 000000 000000 000000
4629 017042 000000
4630 017044 047613 047604
4631 017050 000000
4632
4633
4634
4635
4636
4637
4638 017052 000004
4639 017054 012705 017066
4640 017060 004737 022522
4641
4642 017064 000413
4643
4644 017066
4645 017066 000177 177777 177777
4646 017074 177777
4647 017076 000000 000000 000000
4648 017104 000000
4649 017106 047753 047744
4650 017112 000000

```

```

*****
; TEST 246 TEST OF CLRD INSTR, DATA SET CLRD-4
; * ROUNING MODE, ALL INTERRUPT ENABLES ON
; *****
†ST246: SCOPE
MOV #CLRD4,R5 ; PTR TO TEST DATA SET
JSR PC,@#CLRDT ; GO TEST
BR TST247 ;;
CLRD4: ; TEST DATA SET CLRD-4:
.WORD ALTN,ALTN,ALTN,ALTN ; INITIAL MEM FLOAT NUMBER
.WORD 0,0,0,0 ; EXPECTED FLOAT RESULT
.WORD 047713,047704 ; FPS: BEFORE, AFTER
.WORD 000000 ; FEC AFTER ( 0 = N/A )
*****
; TEST 247 TEST OF CLRD INSTR, DATA SET CLRD-5
; * ROUNING MODE, ALL INTERRUPT ENABLES ON
; *****
†ST247: SCOPE
MOV #CLRD5,R5 ; PTR TO TEST DATA SET
JSR PC,@#CLRDT ; GO TEST
BR TST250 ;;
CLRD5: ; TEST DATA SET CLRD-5:
.WORD M0,0,0,0 ; INITIAL MEM FLOAT NUMBER
.WORD 0,0,0,0 ; EXPECTED FLOAT RESULT
.WORD 047613,047604 ; FPS: BEFORE, AFTER
.WORD 000000 ; FEC AFTER ( 0 = N/A )
*****
; TEST 250 TEST OF CLRD INSTR, DATA SET CLRD-6
; * TRUNCATE MODE, ALL INTERRUPT ENABLES ON
; *****
†ST250: SCOPE
MOV #CLRD6,R5 ; PTR TO TEST DATA SET
JSR PC,@#CLRDT ; GO TEST
BR TST251 ;;
CLRD6: ; TEST DATA SET CLRD-6:
.WORD ZXIMP,M1,M1,M1 ; INITIAL MEM FLOAT NUMBER
.WORD 0,0,0,0 ; EXPECTED FLOAT RESULT
.WORD 047753,047744 ; FPS: BEFORE, AFTER
.WORD 000000 ; FEC AFTER ( 0 = N/A )

```

E08

FPU BASIC INSTR TESTS MACY11 27(1006) 09-FEB-77 09:48 PAGE 97  
DQFPAA.P11 09-FEB-77 09:42 T250 TEST OF CLRD INSTR, DATA SET CLRD-6

4651  
4652  
4653



F08

FPU BASIC INSTR TESTS MACY11 27(1006) 09-FEB-77 09:48 PAGE 98  
DQFPAA.P11 09-FEB-77 09:42

T251 TEST OF TSTF INSTR, DATA SET TSTF-1

4654  
4655  
4656  
4657  
4658 017114 000004  
4659 017116 012705 017130  
4660 017122 004737 022732  
4661  
4662 017126 000407  
4663  
4664 017130  
4665 017130 000000 000000  
4666 017134 000000 000000  
4667 017140 047513 047504  
4668 017144 000000  
4669  
4670  
4671  
4672  
4673  
4674  
4675 017146 000004  
4676 017150 012705 017162  
4677 017154 004737 022732  
4678  
4679 017160 000407  
4680  
4681 017162  
4682 017162 052525 052525  
4683 017166 052525 052525  
4684 017172 047557 047540  
4685 017176 000000  
4686  
4687  
4688  
4689  
4690  
4691  
4692 017200 000004  
4693 017202 012705 017214  
4694 017206 004737 022732  
4695  
4696 017212 000407  
4697  
4698 017214  
4699 017214 125252 125252  
4700 017220 125252 125252  
4701 017224 047407 047410  
4702 017230 000000  
4703  
4704  
4705  
4706  
4707  
4708  
4709 017232 000004

\*\*\*\*\*  
; \*TEST 251 TEST OF TSTF INSTR, DATA SET TSTF-1  
; \* ROUNDING MODE, ALL INTERRUPT ENABLES ON  
; \*\*\*\*\*

†TST251: SCOPE  
MOV #TSTF1,R5 ; PTR TO TEST DATA SET  
JSR PC,@TSTFT ; GO TEST  
BR TST252 ;;

TSTF1: ; TEST DATA SET TSTF-1:  
.WORD 0,0 ; INITIAL MEM FLOAT NUMBER  
.WORD 0,0 ; EXPECTED FLOAT RESULT  
.WORD 047513,047504 ; FPS: BEFORE, AFTER  
.WORD 000000 ; FEC AFTER ( 0 = N/A )

\*\*\*\*\*  
; \*TEST 252 TEST OF TSTF INSTR, DATA SET TSTF-2  
; \* TRUNCATE MODE, ALL INTERRUPT ENABLES ON  
; \*\*\*\*\*

†TST252: SCOPE  
MOV #TSTF2,R5 ; PTR TO TEST DATA SET  
JSR PC,@TSTFT ; GO TEST  
BR TST253 ;;

TSTF2: ; TEST DATA SET TSTF-2:  
.WORD ALTP,ALTP ; INITIAL MEM FLOAT NUMBER  
.WORD ALTP,ALTP ; EXPECTED FLOAT RESULT  
.WORD 047557,047540 ; FPS: BEFORE, AFTER  
.WORD 000000 ; FEC AFTER ( 0 = N/A )

\*\*\*\*\*  
; \*TEST 253 TEST OF TSTF INSTR, DATA SET TSTF-3  
; \* ROUNDING MODE, ALL INTERRUPT ENABLES ON  
; \*\*\*\*\*

†TST253: SCOPE  
MOV #TSTF3,R5 ; PTR TO TEST DATA SET  
JSR PC,@TSTFT ; GO TEST  
BR TST254 ;;

TSTF3: ; TEST DATA SET TSTF-3:  
.WORD ALTN,ALTN ; INITIAL MEM FLOAT NUMBER  
.WORD ALTN,ALTN ; EXPECTED FLOAT RESULT  
.WORD 047407,047410 ; FPS: BEFORE, AFTER  
.WORD 000000 ; FEC AFTER ( 0 = N/A )

\*\*\*\*\*  
; \*TEST 254 TEST OF TSTF INSTR, DATA SET TSTF-4  
; \* TRUNCATE MODE, ALL INTERRUPT ENABLES ON  
; \*\*\*\*\*

†TST254: SCOPE

4710 017234 012705 017246  
4711 017240 004737 022732  
4712  
4713 017244 000407  
4714  
4715 017246  
4716 017246 177777 177777  
4717 017252 177777 177777  
4718 017256 047447 047450  
4719 017262 000000  
4720  
4721  
4722  
4723  
4724  
4725  
4726 017264 000004  
4727 017266 012705 017300  
4728 017272 004737 022732  
4729  
4730 017276 000407  
4731  
4732 017300  
4733 017300 000200 000000  
4734 017304 000200 000000  
4735 017310 047517 047500  
4736 017314 000000  
4737  
4738  
4739  
4740  
4741  
4742  
4743 017316 000004  
4744 017320 012705 017332  
4745 017324 004737 022732  
4746  
4747 017330 000407  
4748  
4749 017332  
4750 017332 077777 177777  
4751 017336 077777 177777  
4752 017342 047557 047540  
4753 017346 000000  
4754  
4755  
4756  
4757  
4758  
4759  
4760 017350 000004  
4761 017352 012705 017364  
4762 017356 004737 022732  
4763  
4764 017362 000407  
4765

MOV #TSTF4,R5 ; PTR TO TEST DATA SET  
JSR PC,@TSTFT ; GO TEST  
BR TST255 ;  
TSTF4: ; TEST DATA SET TSTF-4:  
.WORD LGN,M1 ; INITIAL MEM FLOAT NUMBER  
.WORD LGN,M1 ; EXPECTED FLOAT RESULT  
.WORD 047447,047450 ; FPS: BEFORE, AFTER  
.WORD 000000 ; FEC AFTER ( 0 = N/A )

\*\*\*\*\*  
\*TEST 255 TEST OF TSTF INSTR, DATA SET TSTF-5  
\* ROUNDING MODE, ALL INTERRUPT ENABLES ON  
\*\*\*\*\*

TST255: SCOPE  
MOV #TSTF5,R5 ; PTR TO TEST DATA SET  
JSR PC,@TSTFT ; GO TEST  
BR TST256 ;

TSTF5: ; TEST DATA SET TSTF-5:  
.WORD SMP,0 ; INITIAL MEM FLOAT NUMBER  
.WORD SMP,0 ; EXPECTED FLOAT RESULT  
.WORD 047517,047500 ; FPS: BEFORE, AFTER  
.WORD 000000 ; FEC AFTER ( 0 = N/A )

\*\*\*\*\*  
\*TEST 256 TEST OF TSTF INSTR, DATA SET TSTF-6  
\* TRUNCATE MODE, ALL INTERRUPT ENABLES ON  
\*\*\*\*\*

TST256: SCOPE  
MOV #TSTF6,R5 ; PTR TO TEST DATA SET  
JSR PC,@TSTFT ; GO TEST  
BR TST257 ;

TSTF6: ; TEST DATA SET TSTF-6:  
.WORD LGP,M1 ; INITIAL MEM FLOAT NUMBER  
.WORD LGP,M1 ; EXPECTED FLOAT RESULT  
.WORD 047557,047540 ; FPS: BEFORE, AFTER  
.WORD 000000 ; FEC AFTER ( 0 = N/A )

\*\*\*\*\*  
\*TEST 257 TEST OF TSTF INSTR, DATA SET TSTF-7  
\* ROUNDING MODE, ALL INTERRUPT ENABLES ON  
\*\*\*\*\*

TST257: SCOPE  
MOV #TSTF7,R5 ; PTR TO TEST DATA SET  
JSR PC,@TSTFT ; GO TEST  
BR TST260 ;

4766 017364  
 4767 017364 100200 000000  
 4768 017370 100200 000000  
 4769 017374 047407 047410  
 4770 017400 000000  
 4771  
 4772  
 4773  
 4774  
 4775  
 4776  
 4777 017402 000004  
 4778 017404 012705 017416  
 4779 017410 004737 022732  
 4780  
 4781 017414 000407  
 4782  
 4783 017416  
 4784 017416 000177 177777  
 4785 017422 000177 177777  
 4786 017426 047453 047444  
 4787 017432 000000  
 4788  
 4789  
 4790  
 4791  
 4792  
 4793  
 4794 017434 000004  
 4795 017436 012705 017450  
 4796 017442 004737 022732  
 4797  
 4798 017446 000407  
 4799  
 4800 017450  
 4801 017450 100000 000000  
 4802 017454 100000 000000  
 4803 017460 047503 147514  
 4804 017464 100014  
 4805  
 4806  
 4807  
 4808  
 4809  
 4810  
 4811 017466 000004  
 4812 017470 012705 017502  
 4813 017474 004737 022732  
 4814  
 4815 017500 000407  
 4816  
 4817 017502  
 4818 017502 100000 000001  
 4819 017506 100000 000001  
 4820 017512 047543 147554  
 4821 017516 100014

TSTF7: ; TEST DATA SET TSTF-7:  
 .WORD SMN,0 ; INITIAL MEM FLOAT NUMBER  
 .WORD SMN,0 ; EXPECTED FLOAT RESULT  
 .WORD 047407,047410 ; FPS: BEFORE, AFTER  
 .WORD 000000 ; FEC AFTER ( 0 = N/A )

\*\*\*\*\*  
 ;\*TEST 260 TEST OF TSTF INSTR, DATA SET TSTF-10  
 ;\* TRUNCATE MODE, ALL INTERRUPT ENABLES ON  
 ;\*\*\*\*\*

TST260: SCOPE  
 MOV #TSTF10,R5 ; PTR TO TEST DATA SET  
 JSR PC,@TSTFT ; GO TEST  
 BR TST261 ;;

TSTF10: ; TEST DATA SET TSTF-10:  
 .WORD ZXIMP,M1 ; INITIAL MEM FLOAT NUMBER  
 .WORD ZXIMP,M1 ; EXPECTED FLOAT RESULT  
 .WORD 047453,047444 ; FPS: BEFORE, AFTER  
 .WORD 000000 ; FEC AFTER ( 0 = N/A )

\*\*\*\*\*  
 ;\*TEST 261 TEST OF TSTF INSTR, DATA SET TSTF-11  
 ;\* ROUNDING MODE, ALL INTERRUPT ENABLES ON  
 ;\*\*\*\*\*

TST261: SCOPE  
 MOV #TSTF11,R5 ; PTR TO TEST DATA SET  
 JSR PC,@TSTFT ; GO TEST  
 BR TST262 ;;

TSTF11: ; TEST DATA SET TSTF-11:  
 .WORD MO,0 ; INITIAL MEM FLOAT NUMBER  
 .WORD MO,0 ; EXPECTED FLOAT RESULT  
 .WORD 047503,147514 ; FPS: BEFORE, AFTER  
 .WORD 100014 ; FEC AFTER ( 0 = N/A )

\*\*\*\*\*  
 ;\*TEST 262 TEST OF TSTF INSTR, DATA SET TSTF-12  
 ;\* TRUNCATE MODE, ALL INTERRUPT ENABLES ON  
 ;\*\*\*\*\*

TST262: SCOPE  
 MOV #TSTF12,R5 ; PTR TO TEST DATA SET  
 JSR PC,@TSTFT ; GO TEST  
 BR TST263 ;;

TSTF12: ; TEST DATA SET TSTF-12:  
 .WORD MO,1 ; INITIAL MEM FLOAT NUMBER  
 .WORD MO,1 ; EXPECTED FLOAT RESULT  
 .WORD 047543,147554 ; FPS: BEFORE, AFTER  
 .WORD 100014 ; FEC AFTER ( 0 = N/A )

```

4822
4823
4824
4825
4826
4827
4828 017520 000004
4829 017522 012705 017534
4830 017526 004737 022732
4831
4832 017532 000407
4833
4834 017534
4835 017534 100177 177777
4836 017540 100177 177777
4837 017544 043403 043414
4838 017550 000000
4839
4840
4841

```

```

;*****
;TEST 263 TEST OF TSTF INSTR, DATA SET TSTF-13
;* ROUNDING MODE, -0 INTERRUPT ENABLE OFF, ALL OTHERS ON
;*****
TST263: SCOPE
MOV #TSTF13,R5 ; PTR TO TEST DATA SET
JSR PC,@TSTFT ; GO TEST
BR TST264 ;;

TSTF13: ; TEST DATA SET TSTF-13:
.WORD ZX1MN,M1 ; INITIAL MEM FLOAT NUMBER
.WORD ZX1MN,M1 ; EXPECTED FLOAT RESULT
.WORD 043403,043414 ; FPS: BEFORE, AFTER
.WORD 000000 ; FEC AFTER ( 0 = N/A )

```

J08

FPU BASIC INSTR TESTS MACY11 27(1006) 09-FEB-77 09:48 PAGE 102  
DQFPAA.P11 09-FEB-77 09:42

T264 TEST OF TSTD INSTR, DATA SET TSTD-1

4842  
4843  
4844  
4845  
4846 017552 000004  
4847 017554 012705 017566  
4848 017560 004737 023106  
4849  
4850 017564 000413  
4851  
4852 017566  
4853 017566 000000 000000 000000  
4854 017574 000000  
4855 017576 000000 000000 000000  
4856 017604 000000  
4857 017606 047653 047644  
4858 017612 000000  
4859  
4860  
4861  
4862  
4863  
4864  
4865 017614 000004  
4866 017616 012705 017630  
4867 017622 004737 023106  
4868  
4869 017626 000413  
4870  
4871 017630  
4872 017630 052525 052525 052525  
4873 017636 052525  
4874 017640 052525 052525 052525  
4875 017646 052525  
4876 017650 047717 047700  
4877 017654 000000  
4878  
4879  
4880  
4881  
4882  
4883  
4884 017656 000004  
4885 017660 012705 017672  
4886 017664 004737 023106  
4887  
4888 017670 000413  
4889  
4890 017672  
4891 017672 125252 125252 125252  
4892 017700 125252  
4893 017702 125252 125252 125252  
4894 017710 125252  
4895 017712 047747 047750  
4896 017716 000000  
4897

```

*****
*TEST 264 TEST OF TSTD INSTR, DATA SET TSTD-1
* TRUNCATE MODE, ALL INTERRUPT ENABLES ON
*****

```

```

TST264: SCOPE
MOV #TSTD1,R5 ; PTR TO TEST DATA SET
JSR PC,@TSTD1 ; GO TEST
BR TST265 ;;

```

```

TSTD1: ; TEST DATA SET TSTD-1:
.WORD 0,0,0,0 ; INITIAL MEM FLOAT NUMBER
.WORD 0,0,0,0 ; EXPECTED FLOAT RESULT
.WORD 047653,047644 ; FPS: BEFORE, AFTER
.WORD 000000 ; FEC AFTER ( 0 = N/A )

```

```

*****
*TEST 265 TEST OF TSTD INSTR, DATA SET TSTD-2
* ROUNDING MODE, ALL INTERRUPT ENABLES ON
*****

```

```

TST265: SCOPE
MOV #TSTD2,R5 ; PTR TO TEST DATA SET
JSR PC,@TSTD2 ; GO TEST
BR TST266 ;;

```

```

TSTD2: ; TEST DATA SET TSTD-2:
.WORD ALTP,ALTP,ALTP,ALTP ; INITIAL MEM FLOAT NUMBER
.WORD ALTP,ALTP,ALTP,ALTP ; EXPECTED FLOAT RESULT
.WORD 047717,047700 ; FPS: BEFORE, AFTER
.WORD 000000 ; FEC AFTER ( 0 = N/A )

```

```

*****
*TEST 266 TEST OF TSTD INSTR, DATA SET TSTD-3
* TRUNCATE MODE, ALL INTERRUPT ENABLES ON
*****

```

```

TST266: SCOPE
MOV #TSTD3,R5 ; PTR TO TEST DATA SET
JSR PC,@TSTD3 ; GO TEST
BR TST267 ;;

```

```

TSTD3: ; TEST DATA SET TSTD-3:
.WORD ALTN,ALTN,ALTN,ALTN ; INITIAL MEM FLOAT NUMBER
.WORD ALTN,ALTN,ALTN,ALTN ; EXPECTED FLOAT RESULT
.WORD 047747,047750 ; FPS: BEFORE, AFTER
.WORD 000000 ; FEC AFTER ( 0 = N/A )

```

4898  
4899  
4900  
4901  
4902  
4903 017720 000004  
4904 017722 012705 017734  
4905 017726 004737 023106  
4906  
4907 017732 000413  
4908  
4909 017734  
4910 017734 177777 177777 177777  
4911 017742 177777  
4912 017744 177777 177777 177777  
4913 017752 177777  
4914 017754 047607 047610  
4915 017760 000000  
4916  
4917  
4918  
4919  
4920  
4921  
4922 017762 000004  
4923 017764 012705 017776  
4924 017770 004737 023106  
4925  
4926 017774 000413  
4927  
4928 017776  
4929 017776 000200 000000 000000  
4930 020004 000000  
4931 020006 000200 000000 000000  
4932 020014 000000  
4933 020016 047657 047640  
4934 020022 000000  
4935  
4936  
4937  
4938  
4939  
4940  
4941 020024 000004  
4942 020026 012705 020040  
4943 020032 004737 023106  
4944  
4945 020036 000413  
4946  
4947 020040  
4948 020040 077777 177777 177777  
4949 020046 177777  
4950 020050 077777 177777 177777  
4951 020056 177777  
4952 020060 047717 047700  
4953 020064 000000

```

*****
*TEST 267 TEST OF TSTD INSTR, DATA SET TSTD-4
* ROUNDING MODE, ALL INTERRUPT ENABLES ON
*****

```

```

TSTD267: SCOPE
MOV #TSTD4,R5 ; PTR TO TEST DATA SET
JSR PC,@TSTD7 ; GO TEST
BR TSTD270 ;;

TSTD4: ; TEST DATA SET TSTD-4:
.WORD LGN,M1,M1,M1 ; INITIAL MEM FLOAT NUMBER
.WORD LGN,M1,M1,M1 ; EXPECTED FLOAT RESULT
.WORD 047607,047610 ; FPS: BEFORE, AFTER
.WORD 000000 ; FEC AFTER ( 0 = N/A )

```

```

*****
*TEST 270 TEST OF TSTD INSTR, DATA SET TSTD-5
* TRUNCATE MODE, ALL INTERRUPT ENABLES ON
*****

```

```

TSTD270: SCOPE
MOV #TSTD5,R5 ; PTR TO TEST DATA SET
JSR PC,@TSTD7 ; GO TEST
BR TSTD271 ;;

TSTD5: ; TEST DATA SET TSTD-5:
.WORD SMP,0,0,0 ; INITIAL MEM FLOAT NUMBER
.WORD SMP,0,0,0 ; EXPECTED FLOAT RESULT
.WORD 047657,047640 ; FPS: BEFORE, AFTER
.WORD 000000 ; FEC AFTER ( 0 = N/A )

```

```

*****
*TEST 271 TEST OF TSTD INSTR, DATA SET TSTD-6
* ROUNDING MODE, ALL INTERRUPT ENABLES ON
*****

```

```

TSTD271: SCOPE
MOV #TSTD6,R5 ; PTR TO TEST DATA SET
JSR PC,@TSTD7 ; GO TEST
BR TSTD272 ;;

TSTD6: ; TEST DATA SET TSTD-6:
.WORD LGP,M1,M1,M1 ; INITIAL MEM FLOAT NUMBER
.WORD LGP,M1,M1,M1 ; EXPECTED FLOAT RESULT
.WORD 047717,047700 ; FPS: BEFORE, AFTER
.WORD 000000 ; FEC AFTER ( 0 = N/A )

```

4954  
4955  
4956  
4957  
4958  
4959  
4960  
4961  
4962  
4963  
4964  
4965  
4966  
4967  
4968  
4969  
4970  
4971  
4972  
4973  
4974  
4975  
4976  
4977  
4978  
4979  
4980  
4981  
4982  
4983  
4984  
4985  
4986  
4987  
4988  
4989  
4990  
4991  
4992  
4993  
4994  
4995  
4996  
4997  
4998  
4999  
5000  
5001  
5002  
5003  
5004  
5005  
5006  
5007  
5008  
5009

020066 000004  
020070 012705 020102  
020074 004737 023106  
  
020100 000413  
  
020102  
020102 100200 000000 000000  
020110 000000  
020112 100200 000000 000000  
020120 000000  
020122 047747 047750  
020126 000000  
  
  
  
  
020130 000004  
020132 012705 020144  
020136 004737 023106  
  
020142 000413  
  
020144  
020144 000177 177777 177777  
020152 177777  
020154 000177 177777 177777  
020162 177777  
020164 047613 047604  
020170 000000  
  
  
  
  
020172 000004  
020174 012705 020206  
020200 004737 023106  
  
020204 000413  
  
020206  
020206 100000 000000 000000  
020214 000000  
020216 100000 000000 000000  
020224 000000  
020226 047643 147654

```
*****
*TEST 272 TEST OF TSTD INSTR, DATA SET TSTD-7
* TRUNCATE MODE, ALL INTERRUPT ENABLES ON
*****
TST272: SCOPE
MOV #TSTD7,R5 ; PTR TO TEST DATA SET
JSR PC,@TSTD7 ; GO TEST
BR TST273 ;;

TSTD7: ; TEST DATA SET TSTD-7:
.WORD SMN,0,0,0 ; INITIAL MEM FLOAT NUMBER
.WORD SMN,0,0,0 ; EXPECTED FLOAT RESULT
.WORD 047747,047750 ; FPS: BEFORE, AFTER
.WORD 000000 ; FEC AFTER ( 0 = N/A )

*****
*TEST 273 TEST OF TSTD INSTR, DATA SET TSTD-10
* ROUNDING MODE, ALL INTERRUPT ENABLES ON
*****
TST273: SCOPE
MOV #TSTD10,R5 ; PTR TO TEST DATA SET
JSR PC,@TSTD10 ; GO TEST
BR TST274 ;;

TSTD10: ; TEST DATA SET TSTD-10:
.WORD ZXIMP,M1,M1,M1 ; INITIAL MEM FLOAT NUMBER
.WORD ZXIMP,M1,M1,M1 ; EXPECTED FLOAT RESULT
.WORD 047613,047604 ; FPS: BEFORE, AFTER
.WORD 000000 ; FEC AFTER ( 0 = N/A )

*****
*TEST 274 TEST OF TSTD INSTR, DATA SET TSTD-11
* TRUNCATE MODE, ALL INTERRUPT ENABLES ON
*****
TST274: SCOPE
MOV #TSTD11,R5 ; PTR TO TEST DATA SET
JSR PC,@TSTD11 ; GO TEST
BR TST275 ;;

TSTD11: ; TEST DATA SET TSTD-11:
.WORD M0,0,0,0 ; INITIAL MEM FLOAT NUMBER
.WORD M0,0,0,0 ; EXPECTED FLOAT RESULT
.WORD 047643,147654 ; FPS: BEFORE, AFTER
```

5010 020232 100014 .WORD 100014 ; FEC AFTER ( 0 = N/A )

5011  
5012  
5013  
5014  
5015  
5016  
5017  
5018  
5019  
5020  
5021  
5022  
5023  
5024  
5025  
5026  
5027  
5028  
5029  
5030  
5031  
5032  
5033  
5034  
5035  
5036  
5037  
5038  
5039  
5040  
5041  
5042  
5043  
5044  
5045  
5046  
5047  
5048  
5049  
5050  
5051

```
*****  
:TEST 275 TEST OF TSTD INSTR, DATA SET TSTD-12  
:RNDNG MODE, ALL INTERRUPT ENABLES ON  
*****
```

```
TST275: SCOPE  
MOV #TSTD12,R5 ; PTR TO TEST DATA SET  
JSR PC,@TSTD1 ; GO TEST
```

```
BR TST276 ;;
```

```
TSTD12: ; TEST DATA SET TSTD-12:  
.WORD MO,0,0,1 ; INITIAL MEM FLOAT NUMBER
```

```
.WORD MO,0,0,1 ; EXPECTED FLOAT RESULT
```

```
.WORD 047703,147714 ; FPS: BEFORE, AFTER  
.WORD 100014 ; FEC AFTER ( 0 = N/A )
```

```
*****  
:TEST 276 TEST OF TSTD INSTR, DATA SET TSTD-13  
:TRUNCATE MODE, -0 INTERRUPT ENABLE OFF, ALL OTHERS ON  
*****
```

```
TST276: SCOPE  
MOV #TSTD13,R5 ; PTR TO TEST DATA SET  
JSR PC,@TSTD1 ; GO TEST
```

```
BR TST277 ;;
```

```
TSTD13: ; TEST DATA SET TSTD-13:  
.WORD ZX1MN,M1,M1,M1 ; INITIAL MEM FLOAT NUMBER
```

```
.WORD ZX1MN,M1,M1,M1 ; EXPECTED FLOAT RESULT
```

```
.WORD 043643,043654 ; FPS: BEFORE, AFTER  
.WORD 000000 ; FEC AFTER ( 0 = N/A )
```



5052  
5053  
5054  
5055  
5056  
5057  
5058  
5059  
5060  
5061  
5062  
5063  
5064  
5065  
5066  
5067  
5068  
5069  
5070  
5071  
5072  
5073  
5074  
5075  
5076  
5077  
5078  
5079  
5080  
5081  
5082  
5083  
5084  
5085  
5086  
5087  
5088  
5089  
5090  
5091  
5092  
5093  
5094  
5095  
5096  
5097  
5098  
5099  
5100  
5101  
5102  
5103  
5104  
5105  
5106  
5107

020340 000004  
020342 076600 000022  
020346 032700 000020  
020352 001423  
020354 032777 000002 160564  
020362 001017  
020364 012701 010000  
020370 076600 000144  
020374 030100  
020376 001402  
020400 040100  
020402 000401  
020404 050100 1\$:  
020406 076600 000344 2\$:  
020412 030100  
020414 001002  
020416 000137 003200  
020422 005037 001104  
020426 005037 001102  
020432 005037 001310  
020436 005237 001332  
020442 042737 100000 001332  
020450 005327  
020452 000001  
020454 003021

```

:*****
:SBTTL SUB PASS END CONTROL

TST277: ;FORCE LAST TEST NUMBER
SCOPE ;CHECK FOR TEST ITERATIONS HERE

;IF TEST ONLY EITHER HFP OR WFP, ENTER "EOP" ROUTINE DIRECTLY

; IF IN ALTERNATE HFP/WFP MODE,
; COMPLEMENT FLAG<5> HFP ENABLE BIT,
; ENTER EOP ROUTINE ONLY IF ABOUT TO TEST HFP NEXT,
; TESTING SEQUENCE IS: PASS#1 HFP SUB-PASS
; PASS#1 WFP SUB-PASS
; PASS#2 HFP SUB-PASS
; ...

MED RWHAMI ;GET WHAMI INTO RO
BIT #BIT04,RO ;1=HFP PRESENT, 0=NONE
BEQ SEOP ;EXIT IF NONE

BIT #SW01,SWR ;1=HFP OR WFP TEST ONLY
BNE SEOP ;0=ALTERNATE HFP AND WFP TESTS

MOV #BIT12,R1 ;HFP PRESENT, AND IN ALTERNATE MODE;
MED RFLAG ;SO READ FLAGS
BIT R1,RO ;COMPLEMENT FLAG<5>=BIT12=HFP ENABLE FLAG
BEQ 1$ ;
BIC R1,RO ;CLEAR BIT 12
BR 2$ ;
BIS R1,RO ;SET BIT 12
MED ,WFLAG ;REWRITE FLAGS

BIT R1,RO ;HFP OR WFP NEXT ?
BNE SEOP ;IF HFP AGAIN, START NEW PASS
JMP @#SUBPAS ;IF WFP, NEXT SUBPASS

```

```

;*****
:SBTTL END OF PASS ROUTINE (MODIFIED SYSMAC)

; *INCREMENT THE PASS NUMBER ($PASS)
; *IF SW<10>=0, DING BELL ON PASS END
; *IF THERE'S A MONITOR, GO TO IT
; * ELSE JUMP TO NEWPAS

```

```

SEOP:
CLR SERFLG ;ZERO ERROR COUNT
CLR $STNM ;ZERO TEST NUMBER
CLR $TIMES ;ZERO NUMBER OF ITERATIONS
INC $PASS ;INCREMENT PASS COUNT,
; BUT NEVER LET IN GO NEGATIVE
DEC (PC)+ ;PASS LOOP ?
SEOPCT: .WORD 1
BGT $DOAGN ;YES

```

5108	020456	012737			MOV	(PC)+, 2(PC)+	:	RESTORE COUNTER
5109	020460	000001			\$ENDCT:	.WORD	:	
5110	020462	020452				\$EOPCT	:	
5111	020464	032777	002000	160454		BIT	:	BELL ON PASS END ?
5112	020472	001002				BNE	:	NO
5113	020474	104401	001314			TYPE	:	YES
5114							:	
5115	020500	013700	000042		\$GET42:	MOV	:	GET MONITOR ADDRESS
5116	020504	001405				BEQ	:	NO MONITOR
5117	020506	000005				RESET	:	CLEAR WORLD
5118							:	
5119	020510	004710			\$ENDAD:	JSR	:	GO TO MONITOR
5120	020512	000240				NOP	:	
5121	020514	000240				NOP	:	RESERVED FOR ACT11
5122	020516	000240				NOP	:	
5123							:	
5124	020520	000137	003142		\$DOAGN:	JMP	:	RETURN
5125							:	
5126							:	

```

.SBTTL SUBR TO PERFORM TEST OF LDD/STD
LDADT:
5127
5128
5129 020524
5130 020524 012700 000013      MOV      #13,R0      ; LOAD $TMPD-12
5131 020530 010501              MOV      R5,R1      ; WITH TEST DATA SETS
5132 020532 012702 001230      MOV      #TMPD,R2   ; FOR DISPLAY LATER
5133 020536 012122              MOV      (R1)+,(R2)+
5134 020540 077002              SOB      R0,-2
5135 020542 012737 020550 001114  MOV      #LDADL,$LPERR ; ERROR LOOPING ADDRESS
5136
5137 020550 170165 000020      LDADL:  LDFPS 20(R5) ; INITIAL FPS
5138
5139 020554 004714              JSR      PC,(R4)    ; GO PERFORM LDD/STD SEQUENCE
5140
5141 020556 170237 002330      STFPS   FPS        ; STORE FPS AFTER
5142 020562 170337 002332      STST    FEC        ; STORE FEC/FEA AFTER
5143
5144 020566 023765 002330 000022  CMP      FPS,22(R5) ; CHECK FPS
5145 020574 001401              BEQ     65$        ; FPS IS OK
5146 020576 104077              ERROR   77        ; FPS BAD
5147 020600 005765 000024      65$:    TST      24(R5) ; DOES FEC/FEA APPLY?
5148 020604 100016              BPL     66$        ; NO - SKIP TEST
5149 020606 010437 002344      MOV      R4,EXPFEA ; GET EXPECTED FEA
5150 020612 062737 000004 002344  ADD      #4,EXPFEA ; AND ADJUST
5151 020620 123765 002332 000024  CMPB    FEC,24(R5) ; COMPARE FEC-S
5152 020626 001004              BNE     64$        ; NOT EQUAL
5153 020630 023737 002334 002344  CMP      FEA,EXPFEA ; COMPARE FEA-S
5154 020636 001401              BEQ     66$        ; FEC, FEA OK
5155 020640 104102              64$:    ERROR   102 ; FEC OR FEA ARE BAD
5156 020642
5157
5158 020642 023765 001170 000010  CMP      $REG0,10(R5) ; 1ST WORD OF RESULT CHECK?
5159 020650 001014              BNE     67$        ; NO
5160 020652 023765 001172 000012  CMP      $REG1,12(R5) ; 2ND WORD OF RESULT CHECK?
5161 020660 001010              BNE     67$        ; NO
5162 020662 023765 001174 000014  CMP      $REG2,14(R5) ; 3RD WORD OF RESULT CHECK?
5163 020670 001004              BNE     67$        ; NO
5164 020672 023765 001176 000016  CMP      $REG3,16(R5) ; 4TH WORD OF RESULT CHECK?
5165 020700 001401              BEQ     68$        ; ALL WORDS OK
5166 020702 104104              67$:    ERROR   104 ; NUMBERS NOT EQUAL
5167 020704
5168
5169 020704 000207              RTS      PC        ; RETURN TO TEST CALLER
5170
5171 ; * * * LDD/STD SEQUENCE SUBR * * * * *
5172
5173 020706 172437 002366      LDARDO: LDD     PREVAC,AC0 ; PREV AC0 CONTENTS
5174 020712 172415              LDD     (R5),AC0    ; LOAD AC0
5175 020714 174037 001170      STD     AC0,$REG0   ; STORE AC0
5176 020720 000207              RTS      PC
5177 020722 172537 002366      LDARD1: LDD     PREVAC,AC1 ; PREV AC1 CONTENTS
5178 020726 172515              LDD     (R5),AC1   ; LOAD AC1
5179 020730 174137 001170      STD     AC1,$REG0   ; STORE AC1
5180 020734 000207              RTS      PC
5181 020736 172637 002366      LDARD2: LDD     PREVAC,AC2 ; PREV AC2 CONTENTS
5182 020742 172615              LDD     (R5),AC2   ; LOAD AC2

```



```

.SBTTL SUBR TO PERFORM TEST OF LDD/LDF/STD

LDAFT:
5202 021026 012700 000015 MOV #15,R0 ; LOAD $TMP0-14
5203 021026 010501 000015 MOV R5,R1 ; WITH TEST DATA SETS
5204 021032 010501 000015 MOV #TMP0,R2 ; FOR DISPLAY LATER
5205 021034 012702 001230 MOV (R1)+,(R2)+
5206 021040 012122 000015 SOB R0,-2
5207 021042 077002 000015 MOV #LDAFL,$LPERR ; ERROR LOOPING ADDRESS
5208 021044 012737 021052 001114 LDAFL: SETD ; D MODE FOR INITIAL LOAD
5209 021052 170011 JSR PC,(R4) ; GO PERFORM LDD/LDF/STD SEQUENCE
5210 021054 004714 STFPS FPS ; STORE FPS AFTER
5211 021056 170237 002330 STST FEC ; STORE FEC/FEA AFTER
5212 021062 170337 002332
5213 021066 023765 002330 000026 CMP FPS,26(R5) ; CHECK FPS
5214 021074 001401 BEQ 65$ ; FPS IS OK
5215 021076 104100 ERROR 100 ; FPS BAD
5216 021100 005765 000030 65$: TST 30(R5) ; DOES FEC/FEA APPLY?
5217 021104 100016 BPL 66$ ; NO - SKIP TEST
5218 021106 010437 002344 MOV R4,EXPFEA ; GET EXPECTED FEA
5219 021112 062737 000006 002344 ADD #6,EXPFEA ; AND ADJUST
5220 021120 123765 002332 000030 CMPB FEC,30(R5) ; COMPARE FEC-S
5221 021126 001004 BNE 64$ ; NOT EQUAL
5222 021130 023737 002334 002344 CMP FEA,EXPFEA ; COMPARE FEA-S
5223 021136 001401 BEQ 66$ ; FEC, FEA OK
5224 021140 104103 64$: ERROR 103 ; FEC OR FEA ARE BAD
5225 021142 66$:
5226 021142 023765 001170 000014 CMP $REG0,14(R5) ; 1ST WORD OF RESULT CHECK?
5227 021150 001014 BNE 67$ ; NO
5228 021152 023765 001172 000016 CMP $REG1,16(R5) ; 2ND WORD OF RESULT CHECK?
5229 021160 001010 BNE 67$ ; NO
5230 021162 023765 001174 000020 CMP $REG2,20(R5) ; 3RD WORD OF RESULT CHECK?
5231 021170 001004 BNE 67$ ; NO
5232 021172 023765 001176 000022 CMP $REG3,22(R5) ; 4TH WORD OF RESULT CHECK?
5233 021200 001401 BEQ 68$ ; ALL WORDS OK
5234 021202 104105 67$: ERROR 105 ; NUMBERS NOT EQUAL
5235 021204 68$:
5236 021204 000207 RTS PC ; RETURN TO TEST CALLER
5237 021204 000207 ; * * * LDD/LDF/STD SEQUENCE SUBR * * * * *
5238 021206 172415 LDARFO: LDD (R5),AC0 ; PREV ACO CONTENTS
5239 021210 170165 000024 LDFPS 24(R5) ; INITIAL FPS
5240 021214 172465 000010 LDF 10(R5),AC0 ; LOAD NEW HALF OF ACO
5241 021220 170011 SETD ; D MODE FOR STORE
5242 021222 174037 001170 STD ACO,$REG0 ; STORE ALL OF RESULT
5243 021226 000207 RTS PC
5244 021230 172515 LDARF1: LDD (R5),AC1 ; PREV AC1 CONTENTS
5245 021232 170165 000024 LDFPS 24(R5) ; INITIAL FPS
5246 021236 172565 000010 LDF 10(R5),AC1 ; LOAD NEW HALF OF AC1
5247 021242 170011 SETD ; D MODE FOR STORE
  
```

5258	021244	174137	001170		STD	AC1,\$REGO	:	STORE ALL OF RESULT
5259	021250	000207			RTS	PC	:	
5260	021252	172615		LDARF2:	LDD	(R5),AC2	:	PREV AC2 CONTENTS
5261	021254	170165	000024		LDFPS	24(R5)	:	INITIAL FPS
5262	021260	172665	000010		LDF	10(R5),AC2	:	LOAD NEW HALF OF AC2
5263	021264	170011			SETD		:	D MODE FOR STORE
5264	021266	174237	001170		STD	AC2,\$REGO	:	STORE ALL OF RESULT
5265	021272	000207			RTS	PC	:	
5266	021274	172715		LDARF3:	LDD	(R5),AC3	:	PREV AC3 CONTENTS
5267	021276	170165	000024		LDFPS	24(R5)	:	INITIAL FPS
5268	021302	172765	000010		LDF	10(R5),AC3	:	LOAD NEW HALF OF AC3
5269	021306	170011			SETD		:	D MODE FOR STORE
5270	021310	174337	001170		STD	AC3,\$REGO	:	STORE ALL OF RESULT
5271	021314	000207			RTS	PC	:	
5272	021316	172415		LDARF4:	LDD	(R5),AC0	:	INTERMEDIATE
5273	021320	174004			STD	AC0,AC4	:	PREV AC4 CONTENTS
5274	021322	170165	000024		LDFPS	24(R5)	:	INITIAL FPS
5275	021326	172465	000010		LDF	10(R5),AC0	:	INTERMEDIATE
5276	021332	174004			STF	AC0,AC4	:	LOAD NEW HALF OF AC4
5277	021334	170011			SETD		:	D MODE FOR STORE
5278	021336	172404			LDD	AC4,AC0	:	STORE ALL OF RESULT
5279	021340	174037	001170		STD	AC0,\$REGO	:	INTERMEDIATE
5280	021344	000207			RTS	PC	:	
5281	021346	172515		LDARF5:	LDD	(R5),AC1	:	INTERMEDIATE
5282	021350	174105			STD	AC1,AC5	:	PREV AC5 CONTENTS
5283	021352	170165	000024		LDFPS	24(R5)	:	INITIAL FPS
5284	021356	172565	000010		LDF	10(R5),AC1	:	INTERMEDIATE
5285	021362	174105			STF	AC1,AC5	:	LOAD NEW HALF OF AC5
5286	021364	170011			SETD		:	D MODE FOR STORE
5287	021366	172505			LDD	AC5,AC1	:	STORE ALL OF RESULT
5288	021370	174137	001170		STD	AC1,\$REGO	:	INTERMEDIATE
5289	021374	000207			RTS	PC	:	

```

5290 .SBTTL SUBR TO TEST THE ABSF INSTRUCTION
5291
5292 021376 ABSFT: MOV #7,R0 ; LOAD $TMP0-6
5293 021376 012700 000007 MOV R5,R1 ; WITH TEST DATA SETS
5294 021402 010501 MOV #TMP0,R2 ; FOR DISPLAY LATER
5295 021404 012702 001230 MOV (R1)+(R2)+
5296 021410 012122 MOV SOB RO,-2
5297 021412 077002 SOB
5298 021414 012737 021422 001114 MOV #ABSFL,$LPERR ; ERROR LOOPING ADDRESS
5299
5300 021422 170001 ABSFL: SETF ; F MODE
5301 021424 011537 001170 MOV (R5),$REG0 ; INITIAL FLOAT NUMBER
5302 021430 016537 000002 001172 MOV 2(R5),$REG1 ; INTO OPERATING ROOM
5303 021436 170165 000010 LDFPS 10(R5) ; INITIAL FPS
5304
5305 021442 170637 001170 ABSFI: ABSF $REG0 ; ABS(($REG0))->$REG0
5306
5307 021446 170237 002330 STFPS FPS ; STORE FPS AFTER
5308 021452 170337 002332 STST FEC ; STORE FEC/FEA AFTER
5309
5310 021456 023765 002330 000012 CMP FPS,12(R5) ; CHECK FPS
5311 021464 001401 BEQ 65$ ; FPS IS OK
5312 021466 104076 ERROR 76 ; FPS BAD
5313 021470 005765 000014 65$: TST 14(R5) ; DOES FEC/FEA APPLY?
5314 021474 100014 BPL 66$ ; NO - SKIP TEST
5315 021476 012737 021442 002344 MOV #ABSFI,EXPFEA ; GET EXPECTED FEA
5316 021504 123765 002332 000014 CMPB FEC,14(R5) ; COMPARE FEC-S
5317 021512 001004 BNE 64$ ; NOT EQUAL
5318 021514 023737 002334 002344 CMP FEA,EXPFEA ; COMPARE FEA-S
5319 021522 001401 BEQ 66$ ; FEC, FEA OK
5320 021524 104101 64$: ERROR 101 ; FEC OR FEA ARE BAD
5321 021526 66$:
5322
5323 021526 023765 001170 000004 CMP $REG0,4(R5) ; 1ST WORD OF RESULT CHECK?
5324 021534 001004 BNE 67$ ; NO
5325 021536 023765 001172 000006 CMP $REG1,6(R5) ; 2ND WORD OF RESULT CHECK?
5326 021544 001401 BEQ 68$ ; ALL WORDS OK
5327 021546 104106 67$: ERROR 106 ; NUMBERS NOT EQUAL
5328 021550 68$:
5329
5330 021550 000207 RTS PC ; RETURN TO TEST CALLER
5331

```

\*\*\*\*\*  
: .SBTTL SUBR TO TEST THE ABSD INSTRUCTION

```

5332
5333
5334
5335 021552 ABSDT: MOV #13,R0 ; LOAD $TMP0-12
5336 021552 012700 000013 MOV R5,R1 ; WITH TEST DATA SETS
5337 021556 010501 MOV #TMP0,R2 ; FOR DISPLAY LATER
5338 021560 012702 001230 MOV (R1)+(R2)+
5339 021564 012122 MOV SOB RO,-2
5340 021566 077002 SOB
5341 021570 012737 021576 001114 MOV #ABSDL,$LPERR ; ERROR LOOPING ADDRESS
5342
5343 021576 170011 ABSDL: SETD ; D MODE
5344 021600 011537 001170 MOV (R5),$REG0 ; INITIAL FLOAT NUMBER
5345 021604 016537 000002 001172 MOV 2(R5),$REG1 ; INTO OPERATING ROOM

```





```

5381 .SBTTL SUBR TO TEST THE NEGF INSTRUCTION
5382
5383 021762 NEGFT:
5384 021762 012700 000007 MOV #7,R0 ; LOAD $TMPD-6
5385 021766 010501 MOV R5,R1 ; WITH TEST DATA SETS
5386 021770 012702 001230 MOV #TMPD,R2 ; FOR DISPLAY LATER
5387 021774 012122 MOV (R1)+,(R2)+ ;
5388 021776 077002 SOB RO,-2 ;
5389 022000 012737 022006 001114 MOV #NEGFL,$LPERR ; ERROR LOOPING ADDRESS
5390
5391 022006 170001 NEGFL: SETF ; F MODE
5392 022010 011537 001170 MOV (R5),$REGO ; INITIAL FLOAT NUMBER
5393 022014 016537 000002 001172 MOV 2(R5),$REG1 ; INTO OPERATING ROOM
5394 022022 170165 000010 LDFPS 10(R5) ; INITIAL FPS
5395
5396 022026 170737 001170 NEGFI: NEGF $REGO ; -($REGO)->$REGO
5397
5398 022032 170237 002330 STFPS FPS ; STORE FPS AFTER
5399 022036 170337 002332 STST FEC ; STORE FEC/FEA AFTER
5400
5401 022042 023765 002330 000012 CMP FPS,12(R5) ; CHECK FPS
5402 022050 001401 BEQ 65$ ; FPS IS OK
5403 022052 104076 ERROR 76 ; FPS BAD
5404 022054 005765 000014 65$: TST 14(R5) ; DOES FEC/FEA APPLY?
5405 022060 100014 BPL 66$ ; NO - SKIP TEST
5406 022062 012737 022026 002344 MOV #NEGFI,EXPFEA ; GET EXPECTED FEA
5407 022070 123765 002332 000014 CMPB FEC,14(R5) ; COMPARE FEC-S
5408 022076 001004 BNE 64$ ; NOT EQUAL
5409 022100 023737 002334 002344 CMP FEA,EXPFEA ; COMPARE FEA-S
5410 022106 001401 BEQ 66$ ; FEC, FEA OK
5411 022110 104101 64$: ERROR 101 ; FEC OR FEA ARE BAD
5412 022112 66$:
5413
5414 022112 023765 001170 000004 CMP $REGO,4(R5) ; 1ST WORD OF RESULT CHECK?
5415 022120 001004 BNE 67$ ; NO
5416 022122 023765 001172 000006 CMP $REG1,6(R5) ; 2ND WORD OF RESULT CHECK?
5417 022130 001401 BEQ 68$ ; ALL WORDS OK
5418 022132 104110 67$: ERROR 110 ; NUMBERS NOT EQUAL
5419 022134 68$:
5420
5421 022134 000207 RTS PC ; RETURN TO TEST CALLER
5422
5423 ;*****
5424 .SBTTL SUBR TO TEST THE NEGD INSTRUCTION
5425
5426 022136 NEGDT:
5427 022136 012700 000013 MOV #13,R0 ; LOAD $TMPD-12
5428 022142 010501 MOV R5,R1 ; WITH TEST DATA SETS
5429 022144 012702 001230 MOV #TMPD,R2 ; FOR DISPLAY LATER
5430 022150 012122 MOV (R1)+,(R2)+ ;
5431 022152 077002 SOB RO,-2 ;
5432 022154 012737 022162 001114 MOV #NEGDL,$LPERR ; ERROR LOOPING ADDRESS
5433
5434 022162 170011 NEGDL: SETD ; D MODE
5435 022164 011537 001170 MOV (R5),$REGO ; INITIAL FLOAT NUMBER
5436 022170 016537 000002 001172 MOV 2(R5),$REG1 ; INTO OPERATING ROOM
  
```

```

5437 022176 016537 000004 001174      MOV      4(R5), $REG2      ;
5438 022204 016537 000006 001176      MOV      6(R5), $REG3      ;
5439 022212 170165 000020                LDFPS    20(R5)           ; INITIAL FPS
5440
5441 022216 170737 001170      NEGDI:  NEGDI  $REG0      ; -($REG0)->$REG0
5442
5443 022222 170237 002330      STFPS    FPS              ; STORE FPS AFTER
5444 022226 170337 002332      STST     FEC              ; STORE FEC/FEA AFTER
5445
5446 022232 023765 002330 000022      CMP      FPS, 22(R5)      ; CHECK FPS
5447 022240 001401                BEQ      65$              ; FPS IS OK
5448 022242 104077                ERROR    77              ; FPS BAD
5449 022244 005765 000024      65$:    TST     24(R5)      ; DOES FEC/FEA APPLY?
5450 022250 100014                BPL      66$              ; NO - SKIP TEST
5451 022252 012737 022216 002344      MOV      #NEGDI, EXPFEA   ; GET EXPECTED FEA
5452 022260 123765 002332 000024      CMPB    FEC, 24(R5)      ; COMPARE FEC-S
5453 022266 001004                BNE      64$              ; NOT EQUAL
5454 022270 023737 002334 002344      CMP      FEA, EXPFEA     ; COMPARE FEA-S
5455 022276 001401                BEQ      66$              ; FEC, FEA OK
5456 022300 104102                64$:    ERROR    102      ; FEC OR FEA ARE BAD
5457 022302
5458
5459 022302 023765 001170 000010      CMP      $REG0, 10(R5)   ; 1ST WORD OF RESULT CHECK?
5460 022310 001014                BNE      67$              ; NO
5461 022312 023765 001172 000012      CMP      $REG1, 12(R5)   ; 2ND WORD OF RESULT CHECK?
5462 022320 001010                BNE      67$              ; NO
5463 022322 023765 001174 000014      CMP      $REG2, 14(R5)   ; 3RD WORD OF RESULT CHECK?
5464 022330 001004                BNE      67$              ; NO
5465 022332 023765 001176 000016      CMP      $REG3, 16(R5)   ; 4TH WORD OF RESULT CHECK?
5466 022340 001401                BEQ      68$              ; ALL WORDS OK
5467 022342 104111                67$:    ERROR    111      ; NUMBERS NOT EQUAL
5468 022344
5469
5470 022344 000207                RTS      PC              ; RETURN TO TEST CALLER
5471
    
```

```

5472 .SBTTL SUBR TO TEST THE CLRF INSTRUCTION
5473
5474 022346 CLRFT:
5475 022346 012700 000007 MOV #7,R0 ; LOAD $TMP0-6
5476 022352 010501 MOV R5,R1 ; WITH TEST DATA SETS
5477 022354 012702 001230 MOV #TMP0,R2 ; FOR DISPLAY LATER
5478 022360 012122 MOV (R1)+,(R2)+
5479 022362 077002 SOB RO,-2
5480 022364 012737 022372 001114 MOV #CLRFL,$LPERR ; ERROR LOOPING ADDRESS
5481
5482 022372 170001 CLRFL: SETF ; F MODE
5483 022374 011537 001170 MOV (R5),$REG0 ; INITIAL FLOAT NUMBER
5484 022400 016537 000002 001172 MOV 2(R5),$REG1 ; INTO OPERATING ROOM
5485 022406 170165 000010 LDFPS 10(R5) ; INITIAL FPS
5486
5487 022412 170437 001170 CLRFI: CLRF $REG0 ; 0->$REG0
5488
5489 022416 170237 002330 STFPS FPS ; STORE FPS AFTER
5490 022422 170337 002332 STST FEC ; STORE FEC/FEA AFTER
5491
5492 022426 023765 002330 000012 CMP FPS,12(R5) ; CHECK FPS
5493 022434 001401 BEQ 65$ ; FPS IS OK
5494 022436 104076 ERROR 76 ; FPS BAD
5495 022440 005765 000014 65$: TST 14(R5) ; DOES FEC/FEA APPLY?
5496 022444 100014 BPL 66$ ; NO - SKIP TEST
5497 022446 012737 022412 002344 MOV #CLRFI,EXPFEA ; GET EXPECTED FEA
5498 022454 123765 002332 000014 CMPB FEC,14(R5) ; COMPARE FEC-S
5499 022462 001004 BNE 64$ ; NOT EQUAL
5500 022464 023737 002334 002344 CMP FEA,EXPFEA ; COMPARE FEA-S
5501 022472 001401 BEQ 66$ ; FEC, FEA OK
5502 022474 104101 64$: ERROR 101 ; FEC OR FEA ARE BAD
5503 022476 66$:
5504
5505 022476 023765 001170 000004 CMP $REG0,4(R5) ; 1ST WORD OF RESULT CHECK?
5506 022504 001004 BNE 67$ ; NO
5507 022506 023765 001172 000006 CMP $REG1,6(R5) ; 2ND WORD OF RESULT CHECK?
5508 022514 001401 BEQ 68$ ; ALL WORDS OK
5509 022516 104112 67$: ERROR 112 ; NUMBERS NOT EQUAL
5510 022520 68$:
5511
5512 022520 000207 RTS PC ; RETURN TO TEST CALLER
5513
5514 ;*****
5515 .SBTTL SUBR TO TEST THE CLRD INSTRUCTION
5516
5517 022522 CLRDT:
5518 022522 012700 000013 MOV #13,R0 ; LOAD $TMP0-12
5519 022526 010501 MOV R5,R1 ; WITH TEST DATA SETS
5520 022530 012702 001230 MOV #TMP0,R2 ; FOR DISPLAY LATER
5521 022534 012122 MOV (R1)+,(R2)+
5522 022536 077002 SOB RO,-2
5523 022540 012737 022546 001114 MOV #CLRD, $LPERR ; ERROR LOOPING ADDRESS
5524
5525 022546 170011 CLRDL: SETD ; D MODE
5526 022550 011537 001170 MOV (R5),$REG0 ; INITIAL FLOAT NUMBER
5527 022554 016537 000002 001172 MOV 2(R5),$REG1 ; INTO OPERATING ROOM

```

```

5528 022562 016537 000004 001174      MOV      4(R5), $REG2      ;
5529 022570 016537 000006 001176      MOV      6(R5), $REG3      ;
5530 022576 170165 000020                LDFPS   20(R5)            ; INITIAL FPS
5531
5532 022602 170437 001170                CLRDI:  CLRD   $REG0      ; 0->$REG0
5533
5534 022606 170237 002330                STFPS   FPS              ; STORE FPS AFTER
5535 022612 170337 002332                STST    FEC              ; STORE FEC/FEA AFTER
5536
5537 022616 023765 002330 000022      CMP     FPS, 22(R5)       ; CHECK FPS
5538 022624 001401                BEQ     65$              ; FPS IS OK
5539 022626 104077                ERROR   77              ; FPS BAD
5540 022630 005765 000024      65$:   TST    24(R5)       ; DOES FEC/FEA APPLY?
5541 022634 100014                BPL     66$              ; NO - SKIP TEST
5542 022636 012737 022602 002344      MOV     #CLRDI, EXPFEA   ; GET EXPECTED FEA
5543 022644 123765 002332 000024      CMPB   FEC, 24(R5)      ; COMPARE FEC-S
5544 022652 001004                BNE     64$              ; NOT EQUAL
5545 022654 023737 002334 002344      CMP     FEA, EXPFEA     ; COMPARE FEA-S
5546 022662 001401                BEQ     66$              ; FEC, FEA OK
5547 022664 104102      64$:   ERROR   102        ; FEC OR FEA ARE BAD
5548 022666                66$:
5549
5550 022666 023765 001170 000010      CMP     $REG0, 10(R5)    ; 1ST WORD OF RESULT CHECK?
5551 022674 001014                BNE     67$              ; NO
5552 022676 023765 001172 000012      CMP     $REG1, 12(R5)    ; 2ND WORD OF RESULT CHECK?
5553 022704 001010                BNE     67$              ; NO
5554 022706 023765 001174 000014      CMP     $REG2, 14(R5)    ; 3RD WORD OF RESULT CHECK?
5555 022714 001004                BNE     67$              ; NO
5556 022716 023765 001176 000016      CMP     $REG3, 16(R5)    ; 4TH WORD OF RESULT CHECK?
5557 022724 001401                BEQ     68$              ; ALL WORDS OK
5558 022726 104113      67$:   ERROR   113        ; NUMBERS NOT EQUAL
5559 022730                68$:
5560
5561 022730 000207                RTS     PC                ; RETURN TO TEST CALLER
5562
5563

```

```

5564
5565
5566 022732
5567 022732 012700 000007
5568 022736 010501
5569 022740 012702 001230
5570 022744 012122
5571 022746 077002
5572 022750 012737 022756 001114
5573
5574 022756 170001
5575 022760 011537 001170
5576 022764 016537 000002 001172
5577 022772 170165 000010
5578
5579 022776 170537 001170
5580
5581 023002 170237 002330
5582 023006 170337 002332
5583
5584 023012 023765 002330 000012
5585 023020 001401
5586 023022 104076
5587 023024 005765 000014
5588 023030 100014
5589 023032 012737 022776 002344
5590 023040 123765 002332 000014
5591 023046 001004
5592 023050 023737 002334 002344
5593 023056 001401
5594 023060 104101
5595 023062
5596
5597 023062 023765 001170 000004
5598 023070 001004
5599 023072 023765 001172 000006
5600 023100 001401
5601 023102 104114
5602 023104
5603
5604 023104 000207
5605
5606
5607
5608
5609 023106
5610 023106 012700 000013
5611 023112 010501
5612 023114 012702 001230
5613 023120 012122
5614 023122 077002
5615 023124 012737 023132 001114
5616
5617 023132 170011
5618 023134 011537 001170
5619 023140 016537 000002 001172

.SBTTL SUBR TO TEST THE TSTF INSTRUCTION
TSTFT:
MOV #7,R0 ; LOAD $TMP0-6
MOV R5,R1 ; WITH TEST DATA SETS
MOV #TMP0,R2 ; FOR DISPLAY LATER
MOV (R1)+,(R2)+
SOB RO,-2
MOV #TSTFL,$LPERR ; ERROR LOOPING ADDRESS

TSTFL: SETF ; F MODE
MOV (R5),$REG0 ; INITIAL FLOAT NUMBER
MOV 2(R5),$REG1 ; INTO OPERATING ROOM
LDFPS 10(R5) ; INITIAL FPS

TSTFI: TSTF $REG0 ; ($REG0)->$REG0

STFPS FPS ; STORE FPS AFTER
STST FEC ; STORE FEC/FEA AFTER

CMP FPS,12(R5) ; CHECK FPS
BEQ 65$ ; FPS IS OK
ERROR 76 ; FPS BAD
TST 14(R5) ; DOES FEC/FEA APPLY?
BPL 66$ ; NO - SKIP TEST
MOV #TSTFI,EXPFEA ; GET EXPECTED FEA
CMPB FEC,14(R5) ; COMPARE FEC-S
BNE 64$ ; NOT EQUAL
CMP FEA,EXPFEA ; COMPARE FEA-S
BEQ 66$ ; FEC, FEA OK
ERROR 101 ; FEC OR FEA ARE BAD
64$:
66$:

CMP $REG0,4(R5) ; 1ST WORD OF RESULT CHECK?
BNE 67$ ; NO
CMP $REG1,6(R5) ; 2ND WORD OF RESULT CHECK?
BEQ 68$ ; ALL WORDS OK
ERROR 114 ; NUMBERS NOT EQUAL
67$:
68$:

RTS PC ; RETURN TO TEST CALLER

;*****
.SBTTL SUBR TO TEST THE TSTD INSTRUCTION
TSTD:
MOV #13,R0 ; LOAD $TMP0-12
MOV R5,R1 ; WITH TEST DATA SETS
MOV #TMP0,R2 ; FOR DISPLAY LATER
MOV (R1)+,(R2)+
SOB RO,-2
MOV #TSTD,$LPERR ; ERROR LOOPING ADDRESS

TSDL: SETD ; D MODE
MOV (R5),$REG0 ; INITIAL FLOAT NUMBER
MOV 2(R5),$REG1 ; INTO OPERATING ROOM
  
```



5655			
5656			
5657	023316	010637	002342
5658	023322	012637	002336
5659	023326	012637	002340
5660	023332	170237	002330
5661	023336	170337	002332
5662	023342	104116	
5663	023344	013746	002340
5664	023350	013746	002336
5665	023354	000002	

.SBTTL FPP TRAP CATCHER

FPPILT:	MOV	SP,FPPOSP	:	GET NEW SP
	MOV	(SP)+,FPPOPC	:	POP OLD PC FOR DISPLAY
	MOV	(SP)+,FPPOPS	:	POP OLD PS FOR DISPLAY
	STFPS	FPS	:	GET FPS
	STST	FEC	:	GET FEC/FEA
	ERROR	116	:	SIGNAL UNEXPECTED FPP TRAP
	MOV	FPPOPS,-(SP)	:	PUSH PSW
	MOV	FPPOPC,-(SP)	:	PUSH PC
	RTI		:	CONTINUE, RECOVER AT LAST TRAP ONLY

```

5666 .SBTTL SCOPE HANDLER ROUTINE
5667
5668 ;*****
5669 ;THIS ROUTINE CONTROLS THE LOOPING OF SUBTESTS. IT WILL INCREMENT
5670 ;AND LOAD THE TEST NUMBER($STSTM) INTO THE DISPLAY REG.(DISPLAY<15:0>)
5671 ;THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
5672 ;#SW14=1 LOOP ON TEST
5673 ;#SW11=1 INHIBIT ITERATIONS
5674 ;#SW09=1 LOOP ON ERROR
5675 ;#SW08=1 LOOP ON TEST IN "$LPTST"
5676 ;CALL
5677 ;* SCOPE ;;SCOPE=IOT
5678
5679 023356 $SCOPE:
5680 023356 64$:
5681 023356 032777 040000 155562 1$: BIT #BIT14,$SWR ;;LOOP ON PRESENT TEST?
5682 023364 001114 BNE $OVER ;;YES IF SW14=1
5683 ;*****START OF CODE FOR THE XOR TESTER*****
5684 023366 000416 $XTSTR: BR 6$ ;;IF RUNNING ON THE "XOR" TESTER CHANGE
5685 ;THIS INSTRUCTION TO A "NOP" (NOP=240)
5686 023370 013746 000004 MOV $ERRVEC,-(SP) ;;SAVE THE CONTENTS OF THE ERROR VECTOR
5687 023374 012737 023414 000004 MOV #5,$ERRVEC ;;SET FOR TIMEOUT
5688 023402 005737 177060 TST #177060 ;;TIME OUT ON XOR?
5689 023406 012637 000004 MOV (SP)+,$ERRVEC ;;RESTORE THE ERROR VECTOR
5690 023412 000463 BR $SVLAD ;;GO TO THE NEXT TEST
5691 023414 022626 5$: CMP (SP)+,(SP)+ ;;CLEAR THE STACK AFTER A TIME OUT
5692 023416 012637 000004 MOV (SP)+,$ERRVEC ;;RESTORE THE ERROR VECTOR
5693 023422 000423 BR 7$ ;;LOOP ON THE PRESENT TEST
5694 023424 6$;*****END OF CODE FOR THE XOR TESTER*****
5695 023424 032777 000400 155514 BIT #BIT08,$SWR ;;LOOP ON SPEC. TEST?
5696 023432 001404 BEQ 2$ ;;BR IF NO
5697 023434 023737 001112 001102 CMP $LPTST,$STSTM ;;ON THE RIGHT TEST?
5698 023442 001465 BEQ $OVER ;;BR IF YES
5699 023444 005737 001104 2$: TST $ERFLG ;;HAS AN ERROR OCCURRED?
5700 023450 001421 BEQ 3$ ;;BR IF NO
5701 023452 023737 001122 001104 CMP $ERMAX,$ERFLG ;;MAX. ERRORS FOR THIS TEST OCCURRED?
5702 023460 101015 BHI 3$ ;;BR IF NO
5703 023462 032777 001000 155456 BIT #BIT09,$SWR ;;LOOP ON ERROR?
5704 023470 001404 BEQ 4$ ;;BR IF NO
5705 023472 013737 001114 001110 7$: MOV $LPERR,$LPADR ;;SET LOOP ADDRESS TO LAST SCOPE
5706 023500 000446 BR $OVER
5707 023502 005037 001104 4$: CLR $ERFLG ;;ZERO THE ERROR FLAG
5708 023506 005037 001310 CLR $TIMES ;;CLEAR THE NUMBER OF ITERATIONS TO MAKE
5709 023512 000415 BR 1$ ;;ESCAPE TO THE NEXT TEST
5710 023514 032777 004000 155424 3$: BIT #BIT11,$SWR ;;INHIBIT ITERATIONS?
5711 023522 001011 BNE 1$ ;;BR IF YES
5712 023524 005737 001332 TST $PASS ;;IF FIRST PASS OF PROGRAM
5713 023530 001406 BEQ 1$ ;; INHIBIT ITERATIONS
5714 023532 005237 001106 INC $ICNT ;;INCREMENT ITERATION COUNT
5715 023536 023737 001310 001106 CMP $TIMES,$ICNT ;;CHECK THE NUMBER OF ITERATIONS MADE
5716 023544 002024 BGE $OVER ;;BR IF MORE ITERATION REQUIRED
5717 023546 012737 000001 001106 1$: MOV #1,$ICNT ;;REINITIALIZE THE ITERATION COUNTER
5718 023554 013737 023632 001310 MOV $MXCNT,$TIMES ;;SET NUMBER OF ITERATIONS TO DO
5719 023562 005237 001102 $SVLAD: INC $STSTM ;;COUNT TEST NUMBERS
5720 023566 013737 001102 001330 MOV $STSTM,$STESTN ;;SET TEST NUMBER IN APT MAILBOX
5721 023574 011637 001110 MOV (SP),$LPADR ;;SAVE SCOPE LOOP ADDRESS

```



5722	023600	011637	001114		MOV	(SP), \$LPERR	:: SAVE ERROR LOOP ADDRESS
5723	023604	005037	001312		CLR	\$ESCAPE	:: CLEAR THE ESCAPE FROM ERROR ADDRESS
5724	023610	012737	000001	001122	MOV	#1, \$ERMAX	:: ONLY ALLOW ONE(1) ERROR ON NEXT TEST
5725	023616	013777	001102	155324	\$OVER: MOV	\$TSTNM, @DISPLAY	:: DISPLAY TEST NUMBER
5726	023624	013716	001110		MOV	\$LPADR, (SP)	:: FUDGE RETURN ADDRESS
5727	023630	000002			RTI		:: FIXES PS
5728	023632	003720			\$MXCNT: 2000.		:: MAX. NUMBER OF ITERATIONS

5729  
5730  
5731  
5732  
5733  
5734  
5735  
5736  
5737  
5738  
5739  
5740  
5741  
5742  
5743  
5744  
5745  
5746  
5747  
5748  
5749  
5750  
5751  
5752  
5753  
5754  
5755  
5756  
5757  
5758  
5759  
5760  
5761  
5762  
5763  
5764  
5765  
5766  
5767  
5768  
5769  
5770  
5771  
5772  
5773  
5774  
5775  
5776  
5777  
5778  
5779  
5780  
5781  
5782  
5783  
5784

023634  
023634 010037 002346  
023640 010137 002350  
023644 010237 002352  
023650 010337 002354  
023654 010437 002356  
023660 010537 002360  
023664 010637 002362  
023670 062737 000004 002362  
023676 011637 002364  
023702 005237 001104  
023706 001775  
023710 013777 001102 155232  
023716 032777 002000 155222  
023724 001402  
023726 104401 001314  
023732 005237 001116  
023736 011637 001124  
023742 162737 000002 001124  
023750 117737 155150 001120  
023756 032777 020000 155162  
023764 001004  
023766 004737 024076  
023772 104401 001321  
023776  
023776 122737 000001 001344  
024004 001007  
024006 113737 001120 024020  
024014 004737 024612  
024020 000  
024021 000  
024022 000777  
024024 005777 155116  
024030 100001  
024032 000000  
024034 032777 001000 155104  
024042 001402  
024044 013716 001114  
024050 005737 001312  
024054 001402  
024056 013716 001312  
024062

```

.SBTTL ERROR HANDLER ROUTINE
;*****
;THIS ROUTINE WILL INCREMENT THE ERROR FLAG AND THE ERROR COUNT,
;SAVE THE ERROR ITEM NUMBER AND THE ADDRESS OF THE ERROR CALL
;AND GO TO STYPERR ON ERROR
;THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
;SW15=1 HALT ON ERROR
;SW13=1 INHIBIT ERROR TYPEOUTS
;SW10=1 BELL ON ERROR
;SW09=1 LOOP ON ERROR
;CALL
;* ERROR N ;;ERROR=EMT AND N=ERROR ITEM NUMBER

$ERROR:
MOV R0, EREG0 ; DISPLAY R0
MOV R1, EREG1 ; R1
MOV R2, EREG2 ; R2
MOV R3, EREG3 ; R3
MOV R4, EREG4 ; R4
MOV R5, EREG5 ; R5
MOV R6, EREG6 ; GET R6(SP) BEFORE TRAP
ADD #4, EREG6
MOV (SP), EREG7 ; PC -> ERROR CALL INSTR
INC $ERFLG ; SET THE ERROR FLAG
BEQ 7$ ; DON'T LET THE FLAG GO TO ZERO
MOV $STNM, @DISPLAY ; DISPLAY TEST NUMBER
BIT #BIT10, @SWR ; BELL ON ERROR?
BEQ 1$ ; NO - SKIP
TYPE $BELL ; RING BELL
INC $ERTTL ; COUNT THE NUMBER OF ERRORS
MOV (SP), $ERRPC ; GET ADDRESS OF ERROR INSTRUCTION
SUB #2, $ERRPC
MOVB @ $ERRPC, $ITEMB ; STRIP AND SAVE THE ERROR ITEM CODE
BIT #BIT13, @SWR ; SKIP TYPEOUT IF SET
BNE 20$ ; SKIP TYPEOUTS
JSR PC, $STYPERR ; GO TO USER ERROR ROUTINE
TYPE $CRLF

20$:
CMPB #APTENV, $ENV ; RUNNING IN APT MODE
BNE 2$ ; NO SKIP APT ERROR REPORT
MOVB $ITEMB, 21$ ; SET ITEM NUMBER AS ERROR NUMBER
JSR PC, $ATY4 ; REPORT FATAL ERROR TO APT

21$:
.BYTE 0
.BYTE 0

22$:
BR 22$ ; APT ERROR LOOP
TST @SWR ; HALT ON ERROR
BPL 3$ ; SKIP IF CONTINUE
HALT ; HALT ON ERROR!

3$:
BIT #BIT09, @SWR ; LOOP ON ERROR SWITCH SET?
BEQ 4$ ; BR IF NO
MOV $LPERR, (SP) ; FUDGE RETURN FOR LOOPING
TST $ESCAPE ; CHECK FOR AN ESCAPE ADDRESS
BEQ 5$ ; BR IF NONE
MOV $ESCAPE, (SP) ; FUDGE RETURN ADDRESS FOR ESCAPE
    
```

FPU BASIC INSTR TESTS MACY11 27(1006) 09-FEB-77 09:48 PAGE 124  
DQFPAA.P11 09-FEB-77 09:42 ERROR HANDLER ROUTINE

5785	024062	022737	020510	000042		CMP	#SENDAD, 0#42	::ACT-11 AUTO-ACCEPT?
5786	024070	001001				BNE	6S	:::BRANCH IF NO
5787	024072	000000				HALT		:::YES
5788	024074				6S:			
5789	024074	000002			64S:	RTI		:::RETURN

```

5790
5791
5792
5793
5794
5795
5796
5797
5798
5799
5800
5801
5802
5803
5804 024076
5805 024076 104401
5806 024100 001321
5807 024102 010046
5808 024104 010146
5809 024106 005000
5810 024110 153700 001120
5811 024114 001004
5812
5813 024116 013746 001124
5814 024122 104402
5815 024124 000452
5816 024126 005300
5817 024130 006300
5818 024132 010001
5819 024134 006300
5820 024136 060100
5821 024140 062700 001354
5822 024144 012037 024154
5823 024150 001404
5824 024152 104401
5825 024154 000000
5826 024156 104401 001321
5827 024162 104401 024272
5828 024166 012037 024176
5829 024172 001402
5830 024174 104401
5831 024176 000000
5832 024200 104401 001321
5833 024204 017746 000054
5834 024210 104402
5835 024212 104401 024270
5836 024216 017746 000044
5837 024222 104402
5838 024224 104401 024270
5839 024230 011000
5840 024232 001407
5841 024234 013046
5842 024236 104402
5843 024240 005710
5844 024242 001403
5845 024244 104401 024270

```

```

;*****
.SBTTL ERROR MESSAGE TYPEOUT ROUTINE (MODIFIED SYSMAC)
; *THIS ROUTINE USES THE "ITEM CONTROL BYTE" ($ITEMB) TO DETERMINE WHICH
; *ERROR IS TO BE REPORTED. IT THEN OBTAINS, FROM THE "ERROR TABLE",
; *(SERRTB) THE ERROR MESSAGE, DATA HEADER, AND DATA VALUES TO PRINT.
; *THIS ROUTINE IS IDENTICAL TO THE SYSMAC ROUTINE SERRTYP, EXCEPT THIS
; *ROUTINE PUTS A <HT> BETWEEN OCTAL TYPED DATA VALUES, SO THAT EACH
; *VALUE STARTS AT A HORIZONTAL TAB STOP. ALSO, THE DATA FORMAT
; *POINTER HAS BEEN ELIMINATED FROM THE ERROR VECTOR. THIS ROUTINE
; *ALSO ALWAYS PRINTS $TESTN AND SERRPC AS THE FIRST TWO DATA ELEMENTS
; *(WITH APPROPRIATE HEADERS).

```

```

$TYPERR:
HOTWRM: TYPE .WORD $SCLF ; TYPE "HOT" OR "WARM"
MOV RO,-(SP) ; PTR TO MESSAGE
MOV R1,-(SP) ; SAVE RO
CLR RO ; SAVE R1
BISB @#$ITEMB,RO ; PICKUP ITEM INDEX
BNE 1$ ; IF ITEM NUMBER FROM ERROR 0,
; JUST TYPE PC OF ERROR
MOV $ERRPC,-(SP) ; GET ERROR PC FOR TYPEOUT
TPOC ; TYPE OCTAL, ALL DIGITS
BR 7$ ; EXIT
1$: DEC RO ; ADJUST ERROR # FOR TABLE INDEX
ASL RO ; OF 6 BYTES/ENTRY
MOV RO,R1
ASL RO
ADD R1,RO
ADD #$SERRTB,RO ; FORM TABLE PTR
MOV (RO)+,2$ ; PICKUP "ERROR MESSAGE" PTR
BEQ 3$ ; SKIP TYPEOUT IF NULL
TYPE ; TYPE "ERROR MESSAGE"
; "ERROR MESSAGE" PTR HERE
2$: .WORD 0 ; CR & LF
TYPE , $SCLF ; "TEST # ERR PC" HEADER
3$: TYPE 11$ ; PICKUP "DATA HEADER" PTR
MOV (RO)+,4$ ; SKIP TYPEOUT IF NULL
BEQ 5$ ; TYPE "DATA HEADER"
TYPE ; "DATA HEADER" PTR HERE
4$: .WORD 0 ; CR & LF
5$: TYPE $SCLF ; ($TESTN)
MOV @8$,-(SP) ; OCTAL W/ LEADING ZEROS
TPOC ; <HT>
TYPE 10$ ; ($ERRPC)
MOV @9$,-(SP) ; OCTAL W/ LEADING ZEROS
TPOC ; <HT>
TYPE 10$ ; PICKUP "DATA TABLE" PTR
MOV (RO),RO ; EXIT IF NULL
BEQ 7$ ; SAVE ... FOR TYPEOUT
6$: MOV @2(RO)+,-(SP) ; TYPE OCTAL, ALL DIGITS
TST (RO) ; ANOTHER NUMBER ?
BEQ 7$ ; NO - EXIT
TYPE ,10$ ; TAB BETWEEN ELEMENTS

```

# H10

FPU BASIC INSTR TESTS MACY11 27(1006) 09-FEB-77 09:48 PAGE 126  
DQFPAA.P11 09-FEB-77 09:42 ERROR MESSAGE TYPEOUT ROUTINE (MODIFIED SYSMAC)

```
5846 024250 000771
5847 024252 012601
5848 024254 012600
5849 024256 104401 001321
5850 024262 000207
5851 024264 001330
5852 024266 001124
5853 024270 000011
5854 024272 042524 052123 021440
5855 024300 042411 051122 050040
5856 024306 004503 000
5857 024312

7$: BR 6$ ; LOOP ON DATA TABLE VECTOR
MOV (SP)+,R1 ; RESTORE R1
MOV (SP)+,R0 ; RESTORE R0
TYPE $CRLF ; CR & LF
RTS PC ; RETURN
8$: .WORD $TESTN
9$: .WORD $ERRPC
10$: .ASCIZ <11> ; <HT>
11$: .ASCIZ "TEST # ERR PC"

.EVEN
```

.SBTTL TYPE ROUTINE

```

5858
5859
5860
5861
5862
5863
5864
5865
5866
5867
5868
5869
5870
5871
5872
5873
5874
5875 024312 105737 001165 $TYPE: TSTB $TFPLG ;; IS THERE A TERMINAL?
5876 024316 100002 BPL 1$ ;; BR IF YES
5877 024320 000000 HALT ;; HALT HERE IF NO TERMINAL
5878 024322 000430 BR 3$ ;; LEAVE
5879 024324 010046 1$: MOV RO, -(SP) ;; SAVE RO
5880 024326 017600 000002 MOV 22(SP), RO ;; GET ADDRESS OF ASCIZ STRING
5881 024332 122737 000001 001344 CMPB #APTENV, SENV ;; RUNNING IN APT MODE
5882 024340 001011 BNE 62$ ;; NO GO CHECK FOR APT CONSOLE
5883 024342 132737 000100 001345 BITB #APTPOOL, SENVM ;; SPOOL MESSAGE TO APT
5884 024350 001405 BEQ 62$ ;; NO GO CHECK FOR CONSOLE
5885 024352 010037 024362 MOV RO, 61$ ;; SETUP MESSAGE ADDRESS FOR APT
5886 024356 004737 024602 JSR PC, SATY3 ;; SPOOL MESSAGE TO APT
5887 024362 000000 61$: .WORD 0 ;; MESSAGE ADDRESS
5888 024364 132737 000040 001345 62$: BITB #APTCSUP, SENVM ;; APT CONSOLE SUPPRESSED
5889 024372 001003 BNE 60$ ;; YES, SKIP TYPE OUT
5890 024374 112046 2$: MOVB (RO)+, -(SP) ;; PUSH CHARACTER TO BE TYPED ONTO STACK
5891 024376 001005 BNE 4$ ;; BR IF IT ISN'T THE TERMINATOR
5892 024400 005726 TST (SP)+ ;; IF TERMINATOR POP IT OFF THE STACK
5893 024402 012600 60$: MOV (SP)+, RO ;; RESTORE RO
5894 024404 062716 000002 3$: ADD #2, (SP) ;; ADJUST RETURN PC
5895 024410 000002 RTI ;; RETURN
5896 024412 122716 000011 4$: CMPB #HT, (SP) ;; BRANCH IF <HT>
5897 024416 001430 BEQ 8$ ;; BRANCH IF NOT <CRLF>
5898 024420 122716 000200 CMPB #CRLF, (SP) ;; BRANCH IF NOT <CRLF>
5899 024424 001006 BNE 5$
5900 024426 005726 TST (SP)+ ;; POP <CR><LF> EQUIV
5901 024430 104401 TYPE ;; TYPE A CR AND LF
5902 024432 001321 $CRLF
5903 024434 105037 024570 CLRB $CHARCNT ;; CLEAR CHARACTER COUNT
5904 024440 000755 BR 2$ ;; GET NEXT CHARACTER
5905 024442 004737 024524 5$: JSR PC, $TYPEC ;; GO TYPE THIS CHARACTER
5906 024446 123726 001164 6$: CMPB $FILLC, (SP)+ ;; IS IT TIME FOR FILLER CHARS.?
5907 024452 001350 BNE 2$ ;; IF NO GO GET NEXT CHAR.
5908 024454 013746 001162 MOV $NULL, -(SP) ;; GET # OF FILLER CHARS. NEEDED
5909 AND THE NULL CHAR.
5910 024460 105366 000001 7$: DECB 1(SP) ;; DOES A NULL NEED TO BE TYPED?
5911 024464 002770 BLT 6$ ;; BR IF NO--GO POP THE NULL OFF OF STACK
5912 024466 004737 024524 JSR PC, $TYPEC ;; GO TYPE A NULL
5913 024472 105337 024570 DECB $CHARCNT ;; DO NOT COUNT AS A COUNT

```

```

5914 024476 000770          BR      7$          ;;LOOP
5915
5916          ;HORIZONTAL TAB PROCESSOR
5917
5918 024500 112716 000040      8$:      MOVB      #' (SP)          ;; REPLACE TAB WITH SPACE
5919 024504 004737 024524      9$:      JSR      PC,$TYPEC          ;; TYPE A SPACE
5920 024510 132737 000007 024570      BITB      #7,$CHARCNT          ;; BRANCH IF NOT AT
5921 024516 001372          BNE      9$          ;; TAB STOP
5922 024520 005726          TST      (SP)+          ;; POP SPACE OFF STACK
5923 024522 000724          BR      2$          ;; GET NEXT CHARACTER
5924 024524 105777 154426      $TYPEC: TSTB      @STPS          ;; WAIT UNTIL PRINTER IS READY
5925 024530 100375          BPL      $TYPEC
5926 024532 116677 000002 154420      MOVB      2(SP),@STPB          ;; LOAD CHAR TO BE TYPED INTO DATA REG.
5927 024540 122766 000015 000002      CMPB      #CR,2(SP)          ;; IS CHARACTER A CARRIAGE RETURN?
5928 024546 001003          BNE      1$          ;; BRANCH IF NO
5929 024550 105037 024570          CLRB      $CHARCNT          ;; YES--CLEAR CHARACTER COUNT
5930 024554 000406          BR      $TYPEX          ;; EXIT
5931 024556 122766 000012 000002 1$:      CMPB      #LF,2(SP)          ;; IS CHARACTER A LINE FEED?
5932 024564 001402          BEQ      $TYPEX          ;; BRANCH IF YES
5933 024566 105227          INCB      (PC)+          ;; COUNT THE CHARACTER
5934 024570 000000      $CHARCNT: .WORD 0          ;; CHARACTER COUNT STORAGE
5935 024572 000207      $TYPEX: RTS      PC
5936
  
```

.SBTTL APT COMMUNICATIONS ROUTINE

```

5937
5938
5939
5940 024574 112737 000001 025040 $ATY1: MOVB #1,$FFLG ;; TO REPORT FATAL ERROR
5941 024602 112737 000001 025036 $ATY3: MOVB #1,$MFLG ;; TO TYPE A MESSAGE
5942 024610 000403
5943 024612 112737 000001 025040 $ATY4: MOVB #1,$FFLG ;; TO ONLY REPORT FATAL ERROR
5944 024620 $ATYC:
5945 024620 010046 MOV RO,-(SP) ;; PUSH RO ON STACK
5946 024622 010146 MOV R1,-(SP) ;; PUSH R1 ON STACK
5947 024624 105737 025036 TSTB $MFLG ;; SHOULD TYPE A MESSAGE?
5948 024630 001450 BEQ 5$ ;; IF NOT: BR
5949 024632 122737 000001 001344 CMPB #APTENV,$ENV ;; OPERATING UNDER APT?
5950 024640 001031 BNE 3$ ;; IF NOT: BR
5951 024642 132737 000100 001345 BITB #APTPOOL,$ENVM ;; SHOULD SPOOL MESSAGES?
5952 024650 001425 BEQ 3$ ;; IF NOT: BR
5953 024652 017600 000004 MOV #4(SP),RO ;; GET MESSAGE ADDR.
5954 024656 062766 000002 000004 ADD #2,4(SP) ;; BUMP RETURN ADDR.
5955 024664 005737 001324 1$: TST $MSGTYPE ;; SEE IF DONE W/ LAST XMISSION?
5956 024670 001375 BNE 1$ ;; IF NOT: WAIT
5957 024672 010037 001340 MOV RO,$MSGAD ;; PUT ADDR IN MAILBOX
5958 024676 105720 2$: TSTB (RO)+ ;; FIND END OF MESSAGE
5959 024700 001376 BNE 2$
5960 024702 163700 001340 SUB $MSGAD,RO ;; SUB START OF MESSAGE
5961 024706 006200 ASR RO ;; GET MESSAGE LNTH IN WORDS
5962 024710 010037 001342 MOV RO,$MSGLGT ;; PUT LENGTH IN MAILBOX
5963 024714 012737 000004 001324 MOV #4,$MSGTYPE ;; TELL APT TO TAKE MSG.
5964 024722 000413 BR 5$
5965 024724 017637 000004 024750 3$: MOV #4(SP),4$ ;; PUT MSG ADDR IN JSR LINKAGE
5966 024732 062766 000002 000004 ADD #2,4(SP) ;; BUMP RETURN ADDRESS
5967 024740 013746 177776 MOV 177776,-(SP) ;; PUSH 177776 ON STACK
5968 024744 004737 024312 JSR PC,$TYPE ;; CALL TYPE MACRO
5969 024750 000000 4$: .WORD 0
5970 024752 5$:
5971 024752 105737 025040 10$: TSTB $FFLG ;; SHOULD REPORT FATAL ERROR?
5972 024756 001416 BEQ 12$ ;; IF NOT: BR
5973 024760 005737 001344 TST $ENV ;; RUNNING UNDER APT?
5974 024764 001413 BEQ 12$ ;; IF NOT: BR
5975 024766 005737 001324 11$: TST $MSGTYPE ;; FINISHED LAST MESSAGE?
5976 024772 001375 BNE 11$ ;; IF NOT: WAIT
5977 024774 017637 000004 001326 MOV #4(SP),$FATAL ;; GET ERROR #
5978 025002 062766 000002 000004 ADD #2,4(SP) ;; BUMP RETURN ADDR.
5979 025010 005237 001324 INC $MSGTYPE ;; TELL APT TO TAKE ERROR
5980 025014 105037 025040 12$: CLRB $FFLG ;; CLEAR FATAL FLAG
5981 025020 105037 025037 CLRB $LFLG ;; CLEAR LOG FLAG
5982 025024 105037 025036 CLRB $MFLG ;; CLEAR MESSAGE FLAG
5983 025030 012601 MOV (SP)+,R1 ;; POP STACK INTO R1
5984 025032 012600 MOV (SP)+,RO ;; POP STACK INTO RO
5985 025034 000207 RTS PC ;; RETURN
5986 025036 000 $MFLG: .BYTE 0 ;; MESSG. FLAG
5987 025037 000 $LFLG: .BYTE 0 ;; LOG FLAG
5988 025040 000 $FFLG: .BYTE 0 ;; FATAL FLAG
5989 025042 .EVEN
5990 000200 APTSIZE=200
5991 000001 APTENV=001
5992 000100 APTPOOL=100

```



L10

FPU BASIC INSTR TESTS MACY11 27(1006) 09-FEB-77 09:48 PAGE 130  
DQFPAA.P11 09-FEB-77 09:42 APT COMMUNICATIONS ROUTINE

5993

000040

APTCSUP=040

# M10

FPU BASIC INSTR TESTS MACY11 27(1006) 09-FEB-77 09:48 PAGE 131  
 DQFPAA.P11 09-FEB-77 09:42

BINARY TO OCTAL (ASCII) AND TYPE

.SBTTL BINARY TO OCTAL (ASCII) AND TYPE

```

5994
5995
5996
5997
5998
5999
6000
6001
6002
6003
6004
6005
6006
6007
6008
6009
6010
6011
6012
6013
6014
6015
6016
6017
6018
6019 025042 017646 000000
6020 025046 116637 000001 025265
6021 025054 112637 025267
6022 025060 062716 000002
6023 025064 000406
6024 025066 112737 000001 025265
6025 025074 112737 000006 025267
6026 025102 112737 000005 025264
6027 025110 010346
6028 025112 010446
6029 025114 010546
6030 025116 113704 025267
6031 025122 005404
6032 025124 062704 000006
6033 025130 110437 025266
6034 025134 113704 025265
6035 025140 016605 000012
6036 025144 005003
6037 025146 006105
6038 025150 000404
6039 025152 006105
6040 025154 006105
6041 025156 006105
6042 025160 010503
6043 025162 006103
6044 025164 105337 025266
6045 025170 100016
6046 025172 042703 177770
6047 025176 001002
6048 025200 005704
6049 025202 001403
  
```

```

*****
*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIGIT
*OCTAL (ASCII) NUMBER AND TYPE IT.
*STYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
*CALL:
*      MOV      NUM,-(SP)      ;; NUMBER TO BE TYPED
*      TYPOS    ;; CALL FOR TYPEOUT
*      .BYTE   N              ;; N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
*      .BYTE   M              ;; M=1 OR 0
*                               ;; 1=TYPE LEADING ZEROS
*                               ;; 0=SUPPRESS LEADING ZEROS
*
*STYPON---ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST
*STYPOS OR STYPOC
*CALL:
*      MOV      NUM,-(SP)      ;; NUMBER TO BE TYPED
*      TYPON    ;; CALL FOR TYPEOUT
*
*STYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
*CALL:
*      MOV      NUM,-(SP)      ;; NUMBER TO BE TYPED
*      TYPOC    ;; CALL FOR TYPEOUT
*
*STYPOS: MOV      2(SP),-(SP)    ;; PICKUP THE MODE
*         MOV     1(SP),SOFILL  ;; LOAD ZERO FILL SWITCH
*         MOV     (SP)+,SOMODE+1 ;; NUMBER OF DIGITS TO TYPE
*         ADD    #2,(SP)        ;; ADJUST RETURN ADDRESS
*         BR     STYPON
*STYPOC: MOV     #1,SOFILL      ;; SET THE ZERO FILL SWITCH
*         MOV     #6,SOMODE+1  ;; SET FOR SIX(6) DIGITS
*STYPON: MOV     #5,SOCNT      ;; SET THE ITERATION COUNT
*         MOV    R3,-(SP)      ;; SAVE R3
*         MOV    R4,-(SP)      ;; SAVE R4
*         MOV    R5,-(SP)      ;; SAVE R5
*         MOV     SOMODE+1,R4  ;; GET THE NUMBER OF DIGITS TO TYPE
*         NEG    R4
*         ADD    #6,R4         ;; SUBTRACT IT FOR MAX. ALLOWED
*         MOV     R4,SOMODE    ;; SAVE IT FOR USE
*         MOV     SOFILL,R4    ;; GET THE ZERO FILL SWITCH
*         MOV    12(SP),R5     ;; PICKUP THE INPUT NUMBER
*         CLR    R3           ;; CLEAR THE OUTPUT WORD
1$:      ROL    R5             ;; ROTATE MSB INTO "C"
*         BR    3$           ;; GO DO MSB
2$:      ROL    R5             ;; FORM THIS DIGIT
*
*         ROL    R5
*         ROL    R5
*         MOV    R5,R3
3$:      ROL    R3             ;; GET LSB OF THIS DIGIT
*         DECB  SOMODE        ;; TYPE THIS DIGIT?
*         BPL   7$           ;; BR IF NO
*         BIC   #177770,R3   ;; GET RID OF JUNK
*         BNE   4$           ;; TEST FOR 0
*         TST   R4           ;; SUPPRESS THIS 0?
*         BEQ   5$           ;; BR IF YES
  
```

6050	025204	005204		4\$:	INC	R4	:: DON'T SUPPRESS ANYMORE 0'S
6051	025206	052703	000060		BIS	#'0,R3	:: MAKE THIS DIGIT ASCII
6052	025212	052703	000040	5\$:	BIS	#' R3	:: MAKE ASCII IF NOT ALREADY
6053	025216	110337	025262		MOVB	R3,8\$	:: SAVE FOR TYPING
6054	025222	104401	025262		TYPE	8\$	:: GO TYPE THIS DIGIT
6055	025226	105337	025264	7\$:	DECB	\$OCNT	:: COUNT BY 1
6056	025232	003347			BGT	2\$	:: BR IF MORE TO DO
6057	025234	002402			BLT	6\$	:: BR IF DONE
6058	025236	005204			INC	R4	:: INSURE LAST DIGIT ISN'T A BLANK
6059	025240	000744			BR	2\$	:: GO DO THE LAST DIGIT
6060	025242	012605		6\$:	MOV	(SP)+,R5	:: RESTORE R5
6061	025244	012604			MOV	(SP)+,R4	:: RESTORE R4
6062	025246	012603			MOV	(SP)+,R3	:: RESTORE R3
6063	025250	016666	000002 000004		MOV	2(SP),4(SP)	:: SET THE STACK FOR RETURNING
6064	025256	012616			MOV	(SP)+,(SP)	
6065	025260	000002			RTI		:: RETURN
6066	025262	000		8\$:	.BYTE	0	:: STORAGE FOR ASCII DIGIT
6067	025263	000			.BYTE	0	:: TERMINATOR FOR TYPE ROUTINE
6068	025264	000		\$OCNT:	.BYTE	0	:: OCTAL DIGIT COUNTER
6069	025265	000		\$OFILL:	.BYTE	0	:: ZERO FILL SWITCH
6070	025266	000000		\$OMODE:	.WORD	0	:: NUMBER OF DIGITS TO TYPE

6071  
6072  
6073  
6074  
6075  
6076  
6077  
6078  
6079  
6080  
6081  
6082  
6083  
6084  
6085  
6086  
6087  
6088  
6089  
6090  
6091  
6092  
6093  
6094  
6095  
6096  
6097  
6098  
6099  
6100  
6101  
6102  
6103  
6104  
6105  
6106  
6107

025270 010046  
025272 016600 000002  
025276 005740  
025300 111000  
025302 006300  
025304 016000 025324  
025310 000200

000004 000002

.SBTTL TRAP DECODER

\*\*\*\*\*  
\*THIS ROUTINE WILL PICKUP THE LOWER BYTE OF THE "TRAP" INSTRUCTION  
\*AND USE IT TO INDEX THROUGH THE TRAP TABLE FOR THE STARTING ADDRESS  
\*OF THE DESIRED ROUTINE. THEN USING THE ADDRESS OBTAINED IT WILL  
\*GO TO THAT ROUTINE.

```
$TRAP:  MOV    RO,-(SP)      ;;SAVE RO
        MOV    2(SP),RO    ;;GET TRAP ADDRESS
        TST    -(RO)       ;;BACKUP BY 2
        MOVB   (RO),RO     ;;GET RIGHT BYTE OF TRAP
        ASL    RO          ;;POSITION FOR INDEXING
        MOV    $TRPAD(RO),RO ;;INDEX TO TABLE
        RTS    RO          ;;GO TO ROUTINE
```

;;THIS IS USE TO HANDLE THE "GETPRI" MACRO

```
$TRAP2: MOV    (SP),-(SP)   ;;MOVE THE PC DOWN
        MOV    4(SP),2(SP) ;;MOVE THE PSW DOWN
        RTI                          ;;RESTORE THE PSW
```

.SBTTL TRAP TABLE

\*THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLED  
\*BY THE "TRAP" INSTRUCTION.

	ROUTINE		
\$TRPAD:	.WORD	\$TRAP2	
	\$TYPE	;;CALL=TYPE	TRAP+1(104401) TTY TYPEOUT ROUTINE
	\$TYPOC	;;CALL=TYPOC	TRAP+2(104402) TYPE OCTAL NUMBER (WITH LEADING ZEROS)
	\$TYPOS	;;CALL=TYPOS	TRAP+3(104403) TYPE OCTAL NUMBER (NO LEADING ZEROS)
	\$TYPON	;;CALL=TYPON	TRAP+4(104404) TYPE OCTAL NUMBER (AS PER LAST CALL)

.SBTTL POWER DOWN AND UP ROUTINES

6108  
6109  
6110  
6111  
6112  
6113  
6114  
6115  
6116  
6117  
6118  
6119  
6120  
6121  
6122  
6123  
6124  
6125  
6126  
6127  
6128  
6129  
6130  
6131  
6132  
6133  
6134  
6135  
6136  
6137  
6138  
6139  
6140  
6141  
6142  
6143  
6144  
6145  
6146  
6147  
6148  
6149  
6150  
6151  
6152  
6153  
6154

025336 012737 025510 000024  
025344 012737 000340 000026  
025352 010046  
025354 010146  
025356 010246  
025360 010346  
025362 010446  
025364 010546  
025366 017746 153554  
025372 010637 025514  
025376 012737 025410 000024  
025404 000000  
025406 000776  
  
025410 012737 025510 000024  
025416 013706 025514  
025422 005037 025514  
025426 005237 025514  
025432 001375  
025434 011600  
025436 076600 000226  
025442 012677 153500  
025446 012605  
025450 012604  
025452 012603  
025454 012602  
025456 012601  
025460 012600  
025462 012737 025336 000024  
025470 012737 000340 000026  
025476 104401  
025500 025516  
025502 012716  
025504 002506  
025506 000002  
025510 000000  
025512 000776  
025514 000000  
025516 005015 047520 042527  
025524 000122

```
*****  
:POWER DOWN ROUTINE  
$PWRDN: MOV $SILLUP, @PWRVEC ;; SET FOR FAST UP  
MOV @340, @PWRVEC+2 ;; PRIO:7  
RO, -(SP) ;; PUSH R0 ON STACK  
R1, -(SP) ;; PUSH R1 ON STACK  
R2, -(SP) ;; PUSH R2 ON STACK  
R3, -(SP) ;; PUSH R3 ON STACK  
R4, -(SP) ;; PUSH R4 ON STACK  
R5, -(SP) ;; PUSH R5 ON STACK  
@SWR, -(SP) ;; PUSH @SWR ON STACK  
SP, $SAVR6 ;; SAVE SP  
MOV $PWRUP, @PWRVEC ;; SET UP VECTOR  
HALT  
BR .-2 ;; HANG UP  
  
*****  
:POWER UP ROUTINE  
$PWRUP: MOV $SILLUP, @PWRVEC ;; SET FOR FAST DOWN  
MOV $SAVR6, SP ;; GET SP  
CLR $SAVR6 ;; WAIT LOOP FOR THE TTY  
1$: INC $SAVR6 ;; WAIT FOR THE INC  
BNE 1$ ;; OF WORD  
MOV (SP), RO ;; GET SAVED SWR OFF STACK  
MED 226 ;; RESTORE SWR CONTENTS  
MOV (SP)+, @SWR ;; POP STACK INTO @SWR  
MOV (SP)+, R5 ;; POP STACK INTO R5  
MOV (SP)+, R4 ;; POP STACK INTO R4  
MOV (SP)+, R3 ;; POP STACK INTO R3  
MOV (SP)+, R2 ;; POP STACK INTO R2  
MOV (SP)+, R1 ;; POP STACK INTO R1  
MOV (SP)+, RO ;; POP STACK INTO RO  
MOV $PWRDN, @PWRVEC ;; SET UP THE POWER DOWN VECTOR  
MOV @340, @PWRVEC+2 ;; PRIO:7  
TYPE $POWER ;; REPORT THE POWER FAILURE  
$PWRMG: .WORD $POWER ;; POWER FAIL MESSAGE POINTER  
MOV (PC)+, (SP) ;; RESTART AT START  
$PWRAD: .WORD START ;; RESTART ADDRESS  
RTI  
$ILLUP: HALT ;; THE POWER UP SEQUENCE WAS STARTED  
BR .-2 ;; BEFORE THE POWER DOWN WAS COMPLETE  
$SAVR6: 0 ;; PUT THE SP HERE  
$POWER: .ASCIZ <15><12>"POWER"  
  
.EVEN
```

FPU BASIC INSTR TESTS MACY11 27(1006)  
 DQFPAA.P11 09-FEB-77 09:42

09-FEB-77 09:48 PAGE 135  
 ERROR MESSAGES, DATA HEADERS, DATA VECTORS, ETC

```

. SBTTL ERROR MESSAGES, DATA HEADERS, DATA VECTORS, ETC

; MESSAGE PREFIXES
6158 025526 047510 035124 000040 ASCHOT: .ASCIZ "HOT: "
6159 025534 040527 046522 020072 ASCWRM: .ASCIZ "WARM: "
6160 025542 000

; ERROR MESSAGES HERE
6163 025543 122 041505 044505 EMA: .ASCIZ "RECEIVED FPS IS BAD"
6164 025550 042526 020104 050106
6165 025556 020123 051511 041040
6166 025564 042101 000
6167 025567 122 041505 044505 EMB: .ASCIZ "RECEIVED FEC/FEA IS BAD"
6168 025574 042526 020104 042506
6169 025602 027503 042506 020101
6170 025610 051511 041040 042101
6171 025616 000
6172 025617 101 051502 043050 EMC: .ASCIZ "ABS(F/D) OPERATION - RESULT INCORRECT"
6173 025624 042057 020051 050117
6174 025632 051105 052101 047511
6175 025640 020116 020055 042522
6176 025646 052523 052114 044440
6177 025654 041516 051117 042522
6178 025662 052103 000
6179 025665 125 042516 050130 EMD: .ASCIZ "UNEXPECTED FLOATING POINT TRAP, IGNORED & CONTINUING"
6180 025672 041505 042524 020104
6181 025700 046106 040517 044524
6182 025706 043516 050040 044517
6183 025714 052116 052040 040522
6184 025722 026120 044440 047107
6185 025730 051117 042105 023040
6186 025736 041440 047117 044524
6187 025744 052516 047111 000107
6188 025752 047125 041101 042514 EME: .ASCIZ "UNABLE TO REFERENCE FLOATING ACCUMULATOR"
6189 025760 052040 020117 042522
6190 025766 042506 042522 041516
6191 025774 020105 046106 040517
6192 026002 044524 043516 040440
6193 026010 041503 046525 046125
6194 026016 052101 051117 000
6195 026023 122 050111 046120 EMF: .ASCIZ "RIPPLING A 1 THRU FPS FAILED"
6196 026030 047111 020107 020101
6197 026036 020061 044124 052522
6198 026044 043040 051520 043040
6199 026052 044501 042514 000104
6200 026060 044522 050120 044514 EMG: .ASCIZ "RIPPLING A 0 THRU FPS FAILED"
6201 026066 043516 040440 030040
6202 026074 052040 051110 020125
6203 026102 050106 020123 040506
6204 026110 046111 042105 000
6205 026115 123 052105 020106 EMH: .ASCIZ "SETF FAILED TO CLEAR FPS BIT 7"
6206 026122 040506 046111 042105
6207 026130 052040 020117 046103
6208 026136 040505 020122 050106
6209 026144 020123 044502 020124
6210 026152 000067

```

## E11

FPU BASIC INSTR TESTS MACY11 27(1006) 09-FEB-77 09:48 PAGE 136  
 DQFPAA.P11 09-FEB-77 09:42

ERROR MESSAGES, DATA HEADERS, DATA VECTORS, ETC

6211	026154	042523	042124	043040	EMI:	.ASCIZ	"SETD FAILED TO SET FPS BIT 7"
6212	026162	044501	042514	020104			
6213	026170	047524	051440	052105			
6214	026176	043040	051520	041040			
6215	026204	052111	033440	000			
6216	026211	123	052105	020111	EMJ:	.ASCIZ	"SETI FAILED TO CLEAR FPS BIT 6"
6217	026216	040506	046111	042105			
6218	026224	052040	020117	046103			
6219	026232	040505	020122	050106			
6220	026240	020123	044502	020124			
6221	026246	000066					
6222	026250	042523	046124	043040	EMK:	.ASCIZ	"SETL FAILED TO SET FPS BIT 6"
6223	026256	044501	042514	020104			
6224	026264	047524	051440	052105			
6225	026272	043040	051520	041040			
6226	026300	052111	033040	000			
6227	026305	124	051505	020124	EML:	.ASCII	"TEST OF SRC/DST ADDR MODE FAILED - "
6228	026312	043117	051440	041522			
6229	026320	042057	052123	040440			
6230	026326	042104	020122	047515			
6231	026334	042504	043040	044501			
6232	026342	042514	020104	020055			
6233	026350	042522	052523	052114		.ASCIZ	"RESULT DOESN'T CHECK"
6234	026356	042040	042517	047123			
6235	026364	052047	041440	042510			
6236	026372	045503	000				
6237	026375	124	051505	020124	EMM:	.ASCII	"TEST OF SRC/DST ADDR MODE FAILED - "
6238	026402	043117	051440	041522			
6239	026410	042057	052123	040440			
6240	026416	042104	020122	047515			
6241	026424	042504	043040	044501			
6242	026432	042514	020104	020055			
6243	026440	051123	020103	042101		.ASCIZ	"SRC ADDR REG WRONG INCRE/DECRE"
6244	026446	051104	051040	043505			
6245	026454	053440	047522	043516			
6246	026462	044440	041516	042522			
6247	026470	042057	041505	042522			
6248	026476	000					
6249	026477	124	051505	020124	EMN:	.ASCII	"TEST OF SRC/DST ADDR MODE FAILED - "
6250	026504	043117	051440	041522			
6251	026512	042057	052123	040440			
6252	026520	042104	020122	047515			
6253	026526	042504	043040	044501			
6254	026534	042514	020104	020055			
6255	026542	051504	020124	042101		.ASCIZ	"DST ADDR REG WRONG INCRE/DECRE"
6256	026550	051104	051040	043505			
6257	026556	053440	047522	043516			
6258	026564	044440	041516	042522			
6259	026572	042057	041505	042522			
6260	026600	000					
6261	026601	103	041506	020103	EMO:	.ASCIZ	"CFCC FAILED TO COPY CONDITION CODES CORRECTLY"
6262	026606	040506	046111	042105			
6263	026614	052040	020117	047503			
6264	026622	054520	041440	047117			
6265	026630	044504	044524	047117			
6266	026636	041440	042117	051505			

FPU BASIC INSTR TESTS MACY11 27(1006)  
 DQFPAA.P11 09-FEB-77 09:42

09-FEB-77 09:48 PAGE 137  
 ERROR MESSAGES, DATA HEADERS, DATA VECTORS, ETC

6267	026644	041440	051117	042522	
6268	026652	052103	054514	000	
6269	026657	124	051505	020124	EMP: .ASCII "TEST OF FSRC/FDST D ADDR MODE FAILED - "
6270	026664	043117	043040	051123	
6271	026672	027503	042106	052123	
6272	026700	042040	040440	042104	
6273	026706	020122	047515	042504	
6274	026714	043040	044501	042514	
6275	026722	020104	020055		
6276	026726	042522	052523	052114	.ASCIZ "RESULT DOESN'T CHECK"
6277	026734	042040	042517	047123	
6278	026742	052047	041440	042510	
6279	026750	045503	000		
6280	026753	124	051505	020124	EMQ: .ASCII "TEST OF FSRC/FDST D ADDR MODE FAILED - "
6281	026760	043117	043040	051123	
6282	026766	027503	042106	052123	
6283	026774	042040	040440	042104	
6284	027002	020122	047515	042504	
6285	027010	043040	044501	042514	
6286	027016	020104	020055		
6287	027022	051506	041522	040440	.ASCIZ "FSRC ADDR REG WRONG INCRE/DECRE"
6288	027030	042104	020122	042522	
6289	027036	020107	051127	047117	
6290	027044	020107	047111	051103	
6291	027052	027505	042504	051103	
6292	027060	000105			
6293	027062	042524	052123	047440	EMR: .ASCII "TEST OF FSRC/FDST D ADDR MODE FAILED - "
6294	027070	020106	051506	041522	
6295	027076	043057	051504	020124	
6296	027104	020104	042101	051104	
6297	027112	046440	042117	020105	
6298	027120	040506	046111	042105	
6299	027126	026440	040		
6300	027131	106	051504	020124	.ASCIZ "FDST ADDR REG WRONG INCRE/DECRE"
6301	027136	042101	051104	051040	
6302	027144	043505	053440	047522	
6303	027152	043516	044440	041516	
6304	027160	042522	042057	041505	
6305	027166	042522	000		
6306	027171	124	051505	020124	EMS: .ASCII "TEST OF FSRC/FDST F ADDR MODE FAILED - "
6307	027176	043117	043040	051123	
6308	027204	027503	042106	052123	
6309	027212	043040	040440	042104	
6310	027220	020122	047515	042504	
6311	027226	043040	044501	042514	
6312	027234	020104	020055		
6313	027240	042522	052523	052114	.ASCIZ "RESULT DOESN'T CHECK"
6314	027246	042040	042517	047123	
6315	027254	052047	041440	042510	
6316	027262	045503	000		
6317	027265	124	051505	020124	EMT: .ASCII "TEST OF FSRC/FDST F ADDR MODE FAILED - "
6318	027272	043117	043040	051123	
6319	027300	027503	042106	052123	
6320	027306	043040	040440	042104	
6321	027314	020122	047515	042504	
6322	027322	043040	044501	042514	



## G11

FPU BASIC INSTR TESTS MACY11 27(1006)  
 DQFPAA.P11 09-FEB-77 09:42

09-FEB-77 09:48 PAGE 138  
 ERROR MESSAGES, DATA HEADERS, DATA VECTORS, ETC

6323	027330	020104	020055		
6324	027334	051506	041522	040440	.ASCIZ "FSRC ADDR REG WRONG INCRE/DECRE"
6325	027342	042104	020122	042522	
6326	027350	020107	051127	047117	
6327	027356	020107	047111	051103	
6328	027364	027505	042504	051103	
6329	027372	000105			
6330	027374	042524	052123	047440	EMU: .ASCII "TEST OF FSRC/FDST F ADDR MODE FAILED - "
6331	027402	020106	051506	041522	
6332	027410	043057	051504	020124	
6333	027416	020106	042101	051104	
6334	027424	046440	042117	020105	
6335	027432	040506	046111	042105	
6336	027440	026440	040		
6337	027443	106	051504	020124	.ASCIZ "FDST ADDR REG WRONG INCRE/DECRE"
6338	027450	042101	051104	051040	
6339	027456	043505	053440	047522	
6340	027464	043516	044440	041516	
6341	027472	042522	042057	041505	
6342	027500	042522	000		
6343	027503	114	042104	051457	EMV: .ASCIZ "LDD/STD PATTERN IN FPP AC BAD"
6344	027510	042124	050040	052101	
6345	027516	042524	047122	044440	
6346	027524	020116	050106	020120	
6347	027532	041501	041040	042101	
6348	027540	000			
6349	027541	114	042104	046057	EMW: .ASCIZ "LDD/LDF/STD PATTERN IN FPP AC BAD"
6350	027546	043104	051457	042124	
6351	027554	050040	052101	042524	
6352	027562	047122	044440	020116	
6353	027570	050106	020120	041501	
6354	027576	041040	042101	000	
6355	027603	116	043505	043050	EMX: .ASCIZ "NEG(F/D) OPERATION - RESULT INCORRECT"
6356	027610	042057	020051	050117	
6357	027616	051105	052101	047511	
6358	027624	020116	020055	042522	
6359	027632	052523	052114	044440	
6360	027640	041516	051117	042522	
6361	027646	052103	000		
6362	027651	124	052123	043050	EMY: .ASCIZ "TST(F/D) OPERATION - OPERAND MODIFIED AFTER EXECUTION"
6363	027656	042057	020051	050117	
6364	027664	051105	052101	047511	
6365	027672	020116	020055	050117	
6366	027700	051105	047101	020104	
6367	027706	047515	044504	044506	
6368	027714	042105	040440	052106	
6369	027722	051105	042440	042530	
6370	027730	052503	044524	047117	
6371	027736	000			
6372	027737	103	051114	043050	EMZ: .ASCIZ "CLR(F/D) OPERATION - RESULT INCORRECT"
6373	027744	042057	020051	050117	
6374	027752	051105	052101	047511	
6375	027760	020116	020055	042522	
6376	027766	052523	052114	044440	
6377	027774	041516	051117	042522	
6378	030002	052103	000		

# H11

FPU BASIC INSTR TESTS MACY11 27(1006)  
DQFPAA.P11 09-FEB-77 09:42

09-FEB-77 09:48 PAGE 139  
ERROR MESSAGES, DATA HEADERS, DATA VECTORS, ETC

6379	030005	120	020103	047515	EMAA: .ASCIZ "PC MODE 2 ADDR - INCRE OF +0"
6380	030012	042504	031040	040440	
6381	030020	042104	020122	020055	
6382	030026	047111	051103	020105	
6383	030034	043117	025440	000060	
6384	030042	041520	046440	042117	EMAB: .ASCIZ "PC MODE 2 ADDR - INCRE OF +4, +6, OR +10"
6385	030050	020105	020062	042101	
6386	030056	051104	026440	044440	
6387	030064	041516	042522	047440	
6388	030072	020106	032053	020054	
6389	030100	033053	020054	051117	
6390	030106	025440	030061	000	
6391	030113	120	020103	047515	EMAC: .ASCIZ "PC MODE 2 ADDR - INCRE OF +6 OR +10"
6392	030120	042504	031040	040440	
6393	030126	042104	020122	020055	
6394	030134	047111	051103	020105	
6395	030142	043117	025440	020066	
6396	030150	051117	025440	030061	
6397	030156	000			
6398	030157	120	020103	047515	EMAD: .ASCIZ "PC MODE 2 ADDR - INCRE OF +10"
6399	030164	042504	031040	040440	
6400	030172	042104	020122	020055	
6401	030200	047111	051103	020105	
6402	030206	043117	025440	030061	
6403	030214	000			
6404					
6405					

```

6406 ; DATA HEADERS HERE
6407 030215 114 040517 042504 DHA: .ASCIZ "LOADED STORED"
6408 030222 004504 052123 051117
6409 030230 042105 000
6410 030233 123 047524 042522 DHB: .ASCIZ "STORED"
6411 030240 000104
6412 030242 026455 026455 047514 DHC: .ASCIZ "----LOADED---- ----STORED----"
6413 030250 042101 042105 026455
6414 030256 026455 026411 026455
6415 030264 051455 047524 042522
6416 030272 026504 026455 000055
6417 030300 054105 023520 004504 DHD: .ASCIZ "EXP'D RCV'D"
6418 030306 041522 023526 000104
6419 030314 054105 023520 026504 DHE: .ASCIZ "EXP'D-FEC-RCV'D EXP'D-FEA-RCV'D"
6420 030322 042506 026503 041522
6421 030330 023526 004504 054105
6422 030336 023520 026504 042506
6423 030344 026501 041522 023526
6424 030352 000104
6425 030354 026455 042455 050130 DHF: .ASCIZ "---EXPECTED--- ---RECEIVED---"
6426 030362 041505 042524 026504
6427 030370 026455 026411 026455
6428 030376 042522 042503 053111
6429 030404 042105 026455 000055
6430 030412 026455 026455 026455 DHG: .ASCII "-----EXPECTED-----"
6431 030420 026455 026455 042455
6432 030426 050130 041505 042524
6433 030434 026504 026455 026455
6434 030442 026455 026455 026455
6435 030450 011
6436 030451 055 026455 026455 .ASCIZ "-----RECEIVED-----"
6437 030456 026455 026455 026455
6438 030464 042522 042503 053111
6439 030472 042105 026455 026455
6440 030500 026455 026455 026455
6441 030506 000055
6442 030510 046117 020104 041520 DHH: .ASCIZ "OLD PC OLD PS NEW SP FPS FEC FEA"
6443 030516 047411 042114 050040
6444 030524 004523 042516 020127
6445 030532 050123 020011 050106
6446 030540 004523 043040 041505
6447 030546 020011 042506 000101
6448 030554 047514 042101 042105 DHI: .ASCIZ "LOADED STORED R2"
6449 030562 051411 047524 042522
6450 030570 004504 020040 031122
6451 030576 000
6452 030577 114 040517 042504 DHJ: .ASCIZ "LOADED STORED R3"
6453 030604 004504 052123 051117
6454 030612 042105 020011 051040
6455 030620 000063
6456 030622 047514 042101 042105 DHK: .ASCIZ "LOADED STORED R0 R2"
6457 030630 051411 047524 042522
6458 030636 004504 020040 030122
6459 030644 020011 051040 000062
6460 030652 047514 042101 042105 DHL: .ASCIZ "LOADED STORED $REG2 R0"
6461 030660 051411 047524 042522

```

6462	030666	004504	051044	043505					
6463	030674	004462	020040	030122					
6464	030702	000							
6465	030703	114	040517	042504	DHM:	.ASCIZ	"LOADED STORED \$REG2	R2"	
6466	030710	004504	052123	051117					
6467	030716	042105	022011	042522					
6468	030724	031107	020011	051040					
6469	030732	000062							
6470	030734	047514	042101	042105	DHN:	.ASCIZ	"LOADED STORED \$REG2	R4"	
6471	030742	051411	047524	042522					
6472	030750	004504	051044	043505					
6473	030756	004462	020040	032122					
6474	030764	000							
6475	030765	114	040517	042504	DHO:	.ASCIZ	"LOADED STORED \$REG2	R5"	
6476	030772	004504	052123	051117					
6477	031000	042105	022011	042522					
6478	031006	031107	020011	051040					
6479	031014	000065							
6480	031016	047514	042101	042105	DHP:	.ASCIZ	"LOADED STORED \$REG2	R1 R3"	
6481	031024	051411	047524	042522					
6482	031032	004504	051044	043505					
6483	031040	004462	020040	030522					
6484	031046	020011	051040	000063					
6485	031054	047514	042101	042105	DHQ:	.ASCIZ	"LOADED STORED \$REG2	R3 R5"	
6486	031062	051411	047524	042522					
6487	031070	004504	051044	043505					
6488	031076	004462	020040	031522					
6489	031104	020011	051040	000065					
6490	031112	026455	026455	026455	DHR:	.ASCIZ	"-----STORED-----"		
6491	031120	026455	026455	026455					
6492	031126	052123	051117	042105					
6493	031134	026455	026455	026455					
6494	031142	026455	026455	026455					
6495	031150	000							
6496	031151	055	026455	026455	DHS:	.ASCIZ	"-----STORED-----"	R4"	
6497	031156	026455	026455	026455					
6498	031164	051455	047524	042522					
6499	031172	026504	026455	026455					
6500	031200	026455	026455	026455					
6501	031206	004455	020040	032122					
6502	031214	000							
6503	031215	055	026455	026455	DHT:	.ASCIZ	"-----STORED-----"	R5"	
6504	031222	026455	026455	026455					
6505	031230	051455	047524	042522					
6506	031236	026504	026455	026455					
6507	031244	026455	026455	026455					
6508	031252	004455	020040	032522					
6509	031260	000							
6510	031261	055	026455	026455	DHU:	.ASCII	"-----STORED-----"	RO"	
6511	031266	026455	026455	026455					
6512	031274	051455	047524	042522					
6513	031302	026504	026455	026455					
6514	031310	026455	026455	026455					
6515	031316	004455	020040	030122					
6516	031324	020011	051040	000063	DHV:	.ASCIZ	"-----R3-----"		
6517	031332	026455	026455	026455		.ASCII	"-----STORED-----"	R1"	

6518	031340	026455	026455	026455				
6519	031346	052123	051117	042105				
6520	031354	026455	026455	026455				
6521	031362	026455	026455	026455				
6522	031370	020011	051040	061				
6523	031375	040	004440	031522		.ASCIZ	"	R3"
6524	031402	000						
6525	031403	055	026455	026455	DHW:	.ASCII	"-----STORED-----	\$REG10"
6526	031410	026455	026455	026455				
6527	031416	051455	047524	042522				
6528	031424	026504	026455	026455				
6529	031432	026455	026455	026455				
6530	031440	004455	051044	043505				
6531	031446	030061						
6532	031450	020040	051011	000060	DHX:	.ASCIZ	"	R0"
6533	031456	026455	026455	026455		.ASCII	"-----STORED-----	\$REG10"
6534	031464	026455	026455	026455				
6535	031472	052123	051117	042105				
6536	031500	026455	026455	026455				
6537	031506	026455	026455	026455				
6538	031514	022011	042522	030507				
6539	031522	060						
6540	031523	040	004440	031122		.ASCIZ	"	R2"
6541	031530	000						
6542	031531	055	026455	026455	DHY:	.ASCII	"-----STORED-----	\$REG10"
6543	031536	026455	026455	026455				
6544	031544	051455	047524	042522				
6545	031552	026504	026455	026455				
6546	031560	026455	026455	026455				
6547	031566	004455	051044	043505				
6548	031574	030061						
6549	031576	020040	051011	000064	DHZ:	.ASCIZ	"	R4"
6550	031604	026455	026455	026455		.ASCII	"-----STORED-----	\$REG10"
6551	031612	026455	026455	026455				
6552	031620	052123	051117	042105				
6553	031626	026455	026455	026455				
6554	031634	026455	026455	026455				
6555	031642	022011	042522	030507				
6556	031650	060						
6557	031651	011	051044	043505		.ASCIZ	"	\$REG11 R1"
6558	031656	030461	020011	051040				
6559	031664	000061						
6560	031666	026455	026455	026455	DHAA:	.ASCII	"-----STORED-----	\$REG10"
6561	031674	026455	026455	026455				
6562	031702	052123	051117	042105				
6563	031710	026455	026455	026455				
6564	031716	026455	026455	026455				
6565	031724	022011	042522	030507				
6566	031732	060						
6567	031733	011	051044	043505		.ASCIZ	"	\$REG11 R4 R5"
6568	031740	030461	020011	051040				
6569	031746	004464	020040	032522				
6570	031754	000						
6571	031755	055	026455	046055	DHAB:	.ASCIZ	"-----LOADED-----	-----STORED----- R1"
6572	031762	040517	042504	026504				
6573	031770	026455	004455	026455				

6574	031776	026455	052123	051117				
6575	032004	042105	026455	026455				
6576	032012	020011	051040	000061				
6577	032020	026455	026455	047514	DHAC:	.ASCIZ	"-----LOADED-----	-----STORED----- R3"
6578	032026	042101	042105	026455				
6579	032034	026455	026411	026455				
6580	032042	051455	047524	042522				
6581	032050	026504	026455	004455				
6582	032056	020040	031522	000				
6583	032063	055	026455	046055	DHAD:	.ASCII	"-----LOADED-----	-----STORED----- R2"
6584	032070	040517	042504	026504				
6585	032076	026455	004455	026455				
6586	032104	026455	052123	051117				
6587	032112	042105	026455	026455				
6588	032120	020011	051040	062				
6589	032125	011	020040	032522		.ASCIZ	"	R5"
6590	032132	000						
6591	032133	055	026455	046055	DHAE:	.ASCIZ	"-----LOADED-----	-----STORED----- \$REG4"
6592	032140	040517	042504	026504				
6593	032146	026455	004455	026455				
6594	032154	026455	052123	051117				
6595	032162	042105	026455	026455				
6596	032170	022011	042522	032107				
6597	032176	000						
6598	032177	055	026455	046055	DHAF:	.ASCII	"-----LOADED-----	-----STORED----- \$REG4"
6599	032204	040517	042504	026504				
6600	032212	026455	004455	026455				
6601	032220	026455	052123	051117				
6602	032226	042105	026455	026455				
6603	032234	022011	042522	032107				
6604	032242	020011	051040	004460		.ASCIZ	"	R0 R4"
6605	032250	020040	032122	000				
6606	032255	055	026455	046055	DHAG:	.ASCII	"-----LOADED-----	-----STORED----- \$REG4"
6607	032262	040517	042504	026504				
6608	032270	026455	004455	026455				
6609	032276	026455	052123	051117				
6610	032304	042105	026455	026455				
6611	032312	022011	042522	032107				
6612	032320	020011	051040	004461		.ASCIZ	"	R1 R2"
6613	032326	020040	031122	000				
6614	032333	055	026455	046055	DHAH:	.ASCII	"-----LOADED-----	-----STORED----- \$REG4"
6615	032340	040517	042504	026504				
6616	032346	026455	004455	026455				
6617	032354	026455	052123	051117				
6618	032362	042105	026455	026455				
6619	032370	022011	042522	032107				
6620	032376	020011	051040	004464		.ASCIZ	"	R4 R5"
6621	032404	020040	032522	000				
6622	032411	055	026455	046055	DHAI:	.ASCII	"-----LOADED-----	-----STORED----- \$REG4"
6623	032416	040517	042504	026504				
6624	032424	026455	004455	026455				
6625	032432	026455	052123	051117				
6626	032440	042105	026455	026455				
6627	032446	022011	042522	032107				
6628	032454	022011	042522	032507		.ASCIZ	"	\$REG5 R5"
6629	032462	020011	051040	000065				

M11

FPU BASIC INSTR TESTS MACY11 27(1006)  
DQFPAA.P11 09-FEB-77 09:42

09-FEB-77 09:48 PAGE 144  
ERROR MESSAGES, DATA HEADERS, DATA VECTORS, ETC

6630	032470	026455	026455	047514
6631	032476	042101	042105	026455
6632	032504	026455	026411	026455
6633	032512	051455	047524	042522
6634	032520	026504	026455	004455
6635	032526	051044	043505	064
6636	032533	011	051044	043505
6637	032540	004465	020040	030122
6638	032546	020011	051040	000062
6639	032554	026455	026455	047514
6640	032562	042101	042105	026455
6641	032570	026455	026411	026455
6642	032576	051455	047524	042522
6643	032604	026504	026455	000055
6644	032612	000040		
6645				
6646				

DHAJ: .ASCII "----LOADED---- ----STORED---- \$REG4"

.ASCIZ " \$REG5 R0 R2"

DHAK: .ASCIZ "----LOADED---- ----STORED----"

DHAL: .ASCIZ " "

					; DATA ADDRESS VECTOR	
6647						.EVEN
6648						.WORD
6649	032614	001170	000000		DTA:	\$REG0,0
6650	032620	001170	001172	000000	DTB:	\$REG0,\$REG1,0
6651	032626	001170	001172	001174	DTC:	\$REG0,\$REG1,\$REG2,\$REG3,0
6652	032634	001176	000000			
6653	032640	001242	002330	000000	DTD:	\$TMP5,FPS,0
6654	032646	001252	002330	000000	DTE:	\$TMP11,FPS,0
6655	032654	001256	002330	000000	DTF:	\$TMP13,FPS,0
6656	032662	001244	002332	002344	DTG:	\$TMP6,FEC,EXPFEA,FEA,0
6657	032670	002334	000000			
6658	032674	001254	002332	002344	DTH:	\$TMP12,FEC,EXPFEA,FEA,0
6659	032702	002334	000000			
6660	032706	001260	002332	002344	DTI:	\$TMP14,FEC,EXPFEA,FEA,0
6661	032714	002334	000000			
6662	032720	001234	001236	001170	DTJ:	\$TMP2,\$TMP3,\$REG0,\$REG1,0
6663	032726	001172	000000			
6664	032732	001240	001242	001244	DTK:	\$TMP4,\$TMP5,\$TMP6,\$TMP7
6665	032740	001246				
6666	032742	001170	001172	001174		.WORD \$REG0,\$REG1,\$REG2,\$REG3,0
6667	032750	001176	000000			
6668	032754	001244	001246	001250	DTL:	\$TMP6,\$TMP7,\$TMP10,\$TMP11
6669	032762	001252				
6670	032764	001170	001172	001174		.WORD \$REG0,\$REG1,\$REG2,\$REG3,0
6671	032772	001176	000000			
6672	032776	002336	002340	002342	DTM:	\$FPOPC,\$FPOPS,\$FPOSP,\$FPS,\$FEC,\$FEA,0
6673	033004	002330	002332	002334		
6674	033012	000000				
6675	033014	001170	001172	002352	DTN:	\$REG0,\$REG1,\$REG2,0
6676	033022	000000				
6677	033024	001170	001172	002354	DTO:	\$REG0,\$REG1,\$REG3,0
6678	033032	000000				
6679	033034	001170	001172	002346	DTP:	\$REG0,\$REG1,\$REG0,\$REG2,0
6680	033042	002352	000000			
6681	033046	001170	001172	001174	DTQ:	\$REG0,\$REG1,\$REG2,\$REG0,0
6682	033054	002346	000000			
6683	033060	001170	001172	001174	DTR:	\$REG0,\$REG1,\$REG2,\$REG2,0
6684	033066	002352	000000			
6685	033072	001170	001172	001174	DTS:	\$REG0,\$REG1,\$REG2,\$REG4,0
6686	033100	002356	000000			
6687	033104	001170	001172	001174	DTT:	\$REG0,\$REG1,\$REG2,\$REG5,0
6688	033112	002360	000000			
6689	033116	001170	001172	001174	DTU:	\$REG0,\$REG1,\$REG2,\$REG1,\$REG3,0
6690	033124	002350	002354	000000		
6691	033132	001170	001172	001174	DTV:	\$REG0,\$REG1,\$REG2,\$REG3,\$REG5,0
6692	033140	002354	002360	000000		
6693	033146	001200	001202	001204	DTW:	\$REG4,\$REG5,\$REG6,\$REG7,0
6694	033154	001206	000000			
6695	033160	001200	001202	001204	DTX:	\$REG4,\$REG5,\$REG6,\$REG7,\$REG4,0
6696	033166	001206	002356	000000		
6697	033174	001200	001202	001204	DTY:	\$REG4,\$REG5,\$REG6,\$REG7,\$REG5,0
6698	033202	001206	002360	000000		
6699	033210	001200	001202	001204	DTZ:	\$REG4,\$REG5,\$REG6,\$REG7,\$REG0,\$REG3,0
6700	033216	001206	002346	002354		
6701	033224	000000				
6702	033226	001200	001202	001204	DTAA:	\$REG4,\$REG5,\$REG6,\$REG7,\$REG1,\$REG3,0



FPU BASIC INSTR TESTS  
DGFPA.P11 09-FEB-77MACY11 27(1006)  
09:4209-FEB-77 09:48 PAGE 146  
ERROR MESSAGES, DATA HEADERS, DATA VECTORS, ETC

6703	033234	001206	002350	002354				
6704	033242	000000						
6705	033244	001200	001202	001204	DTAB:	.WORD	\$REG4,\$REG5,\$REG6,\$REG7,\$REG10,EREG0,0	
6706	033252	001206	001210	002346				
6707	033260	000000						
6708	033262	001200	001202	001204	DTAC:	.WORD	\$REG4,\$REG5,\$REG6,\$REG7,\$REG10,EREG2,0	
6709	033270	001206	001210	002352				
6710	033276	000000						
6711	033300	001200	001202	001204	DTAD:	.WORD	\$REG4,\$REG5,\$REG6,\$REG7,\$REG10,EREG4,0	
6712	033306	001206	001210	002356				
6713	033314	000000						
6714	033316	001200	001202	001204	DTAE:	.WORD	\$REG4,\$REG5,\$REG6,\$REG7	
6715	033324	001206						
6716	033326	001210	001212	002350		.WORD	\$REG10,\$REG11,EREG1,0	
6717	033334	000000						
6718	033336	001200	001202	001204	DTAF:	.WORD	\$REG4,\$REG5,\$REG6,\$REG7	
6719	033344	001206						
6720	033346	001210	001212	002356		.WORD	\$REG10,\$REG11,EREG4,EREG5,0	
6721	033354	002360	000000					
6722	033360	001170	001172	001174	DTAG:	.WORD	\$REG0,\$REG1,\$REG2,\$REG3,EREG1,0	
6723	033366	001176	002350	000000				
6724	033374	001170	001172	001174	DTAH:	.WORD	\$REG0,\$REG1,\$REG2,\$REG3,EREG3,0	
6725	033402	001176	002354	000000				
6726	033410	001170	001172	001174	DTAI:	.WORD	\$REG0,\$REG1,\$REG2,\$REG3,EREG2,EREG5,0	
6727	033416	001176	002352	002360				
6728	033424	000000						
6729	033426	001170	001172	001174	DTAJ:	.WORD	\$REG0,\$REG1,\$REG2,\$REG3,\$REG4,0	
6730	033434	001176	001200	000000				
6731	033442	001170	001172	001174	DTAK:	.WORD	\$REG0,\$REG1,\$REG2,\$REG3	
6732	033450	001176						
6733	033452	001200	002346	002356		.WORD	\$REG4,EREG0,EREG4,0	
6734	033460	000000						
6735	033462	001170	001172	001174	DTAL:	.WORD	\$REG0,\$REG1,\$REG2,\$REG3	
6736	033470	001176						
6737	033472	001200	002350	002352		.WORD	\$REG4,EREG1,EREG2,0	
6738	033500	000000						
6739	033502	001170	001172	001174	DTAM:	.WORD	\$REG0,\$REG1,\$REG2,\$REG3	
6740	033510	001176						
6741	033512	001200	002356	002360		.WORD	\$REG4,EREG4,EREG5,0	
6742	033520	000000						
6743	033522	001170	001172	001174	DTAN:	.WORD	\$REG0,\$REG1,\$REG2,\$REG3	
6744	033530	001176						
6745	033532	001200	001202	002360		.WORD	\$REG4,\$REG5,EREG5,0	
6746	033540	000000						
6747	033542	001170	001172	001174	DTAO:	.WORD	\$REG0,\$REG1,\$REG2,\$REG3	
6748	033550	001176						
6749	033552	001200	001202	002346		.WORD	\$REG4,\$REG5,EREG0,EREG2,0	
6750	033560	002352	000000					
6751	033564	000000			DTAP:	.WORD	0	
6752								
6753								
6754								
6755		000001						

; THE END  
.END





APASS = 000000	1086	1091							
APRIOR= 000000	1086								
APTCSU= 000040	5888	5993#							
APTENV= 000001	5768	5881	5949	5991#					
APTSIZ= 000200	1309	5990#							
APTSP0= 000100	5883	5951	5992#						
ASCHOT 025526	1390	6158#							
ASCHRM 025534	1392	6159#							
ASWREG= 000000	1086	1099							
ATESTN= 000000	1086	1090							
AUNIT = 000000	1086	1093							
AUSWR = 000000	1086	1100							
AVECT1= 000000	1086								
AVECT2= 000000	1086								
BGNMES 002376	1247#	1318							
BIT0 = 000001	889#								
BIT00 = 000001	879#	889	1398	1420					
BIT01 = 000002	878#	888							
BIT02 = 000004	877#	887							
BIT03 = 000010	876#	886							
BIT04 = 000020	875#	885	1324	1372	1402	1424	5069		
BIT05 = 000040	874#	884	1832						
BIT06 = 000100	873#	883	1473	1489					
BIT07 = 000200	872#	882	1441	1457					
BIT08 = 000400	871#	881	5695						
BIT09 = 001000	870#	880	5703	5778					
BIT1 = 000002	888#								
BIT10 = 002000	869#	5756							
BIT11 = 004000	868#	5710							
BIT12 = 010000	867#	1356	1382	1384	1387	1404	1426	5075	
BIT13 = 020000	866#	1404	1426	5763					
BIT14 = 040000	865#	5681							
BIT15 = 100000	864#								
BIT2 = 000004	887#								
BIT3 = 000010	886#								
BIT4 = 000020	885#								
BIT5 = 000040	884#								
BIT6 = 000100	883#								
BIT7 = 000200	882#								
BIT8 = 000400	881#								
BIT9 = 001000	880#								
BPTVEC= 000014	896#								
CCONLY= 177760	945#	1830	1839						
CLRD1 022602	5532#	5542							
CLRD1 022546	5523	5525#							
CLRD1 022522	4545	4564	4583	4602	4621	4640	5517#		
CLRD1 016614	4544	4549#							
CLRD2 016656	4563	4568#							
CLRD3 016720	4582	4587#							
CLRD4 016762	4601	4606#							
CLRD5 017024	4620	4625#							
CLRD6 017066	4639	4644#							
CLRF1 022412	5487#	5497							
CLRF1 022372	5480	5482#							
CLRF1 022346	4442	4459	4476	4493	4510	4527	5474#		
CLRF1 016360	4441	4446#							

CLRF2	016412	4458	4463#						
CLRF3	016444	4475	4480#						
CLRF4	016476	4492	4497#						
CLRF5	016530	4509	4514#						
CLRF6	016562	4526	4531#						
CR	= 000015	804#	5927	5937					
CRLF	= 000200	805#	5898	5937					
DDISP	= 177570	811#	1025	1297					
DHA	030215	1129	1130	1154	6407#				
DHAA	031666	1159	1160	6560#					
DHAB	031755	1192	6571#						
DHAC	032020	1183	1184	1186	6577#				
DHAD	032063	1175	1176	6583#					
DHAE	032133	1190	6591#						
DHAF	032177	1177	1178	6598#					
DHAG	032255	1182	6606#						
DHAH	032333	1187	1188	1189	6614#				
DHAI	032411	1185	6622#						
DHAJ	032470	1179	1180	1181	6630#				
DHAK	032554	1191	6639#						
DHAL	032612	1217	1218	1219	1220	6644#			
DHB	030233	1131	1132	1133	1134	6410#			
DHC	030242	1194	6412#						
DHD	030300	1196	1197	1198	6417#				
DHE	030314	1200	1201	1202	6419#				
DHF	030354	1206	1208	1210	1212	6425#			
DHG	030412	1204	1205	1207	1209	1211	1213	6430#	
DHH	030510	1215	6442#						
DHI	030554	1151	6448#						
DHJ	030577	1149	1150	6452#					
DHK	030622	1144	1145	6456#					
DHL	030652	1142	1143	6460#					
DHM	030703	1148	6465#						
DHN	030734	1146	1147	1152	6470#				
DHO	030765	1136	6475#						
DHP	031016	1139	1140	1141	6480#				
DHQ	031054	1137	1138	6485#					
DHR	031112	1172	6490#						
DHS	031151	1157	1158	6496#					
DHT	031215	1156	1161	6503#					
DHU	031261	1162	1163	6510#					
DHV	031332	1170	1171	6517#					
DHW	031403	1166	6525#						
DHX	031456	1168	1169	1173	6533#				
DHY	031531	1167	6542#						
DHZ	031604	1164	1165	6550#					
DISPLA	001150	1025#	1297#	1305#	5725*	5755*			
DISPRE	000174	957#	1305						
DSWR	= 177570	810#	1024	1296					
DTA	032614	1131	1132	1133	1134	6649#			
DTAA	033226	1170	1171	6702#					
DTAB	033244	1166	6705#						
DTAC	033262	1168	1169	1173	6708#				
DTAD	033300	1167	6711#						
DTAE	033316	1164	1165	6714#					
DTAF	033336	1159	1160	6718#					















FPU BASIC INSTR TESTS MACY11 27(1006)  
 DQFPAA.P11 09-FEB-77 09:42

09-FEB-77 09:48 PAGE 158  
 CROSS REFERENCE TABLE -- USER SYMBOLS

TSTF4	017246	4710	4715#
TSTF5	017300	4727	4732#
TSTF6	017332	4744	4749#
TSTF7	017364	4761	4766#
TST1	003304	1397#	
TST10	003652	1516	1524#
TST100	006740	2460	2467#
TST101	006760	2472	2479#
TST102	007000	2484	2492#
TST103	007020	2497	2504#
TST104	007040	2509	2516#
TST105	007060	2521	2528#
TST106	007100	2533	2540#
TST107	007120	2545	2552#
TST11	003732	1544	1552#
TST110	007140	2557	2565#
TST111	007160	2570	2577#
TST112	007200	2582	2589#
TST113	007220	2594	2601#
TST114	007240	2606	2613#
TST115	007260	2618	2625#
TST116	007300	2630	2638#
TST117	007320	2643	2650#
TST12	004022	1577	1585#
TST120	007340	2655	2662#
TST121	007360	2667	2674#
TST122	007400	2679	2686#
TST123	007420	2691	2699#
TST124	007440	2704	2711#
TST125	007460	2716	2723#
TST126	007500	2728	2735#
TST127	007520	2740	2747#
TST13	004100	1604	1612#
TST130	007540	2752	2766#
TST131	007664	2789	2798#
TST132	010020	2826	2835#
TST133	010174	2866	2875#
TST134	010330	2903	2912#
TST135	010466	2941	2950#
TST136	010636	2980	2989#
TST137	011010	3019	3028#
TST14	004152	1631	1639#
TST140	011144	3052	3061#
TST141	011302	3091	3100#
TST142	011442	3129	3138#
TST143	011566	3161	3170#
TST144	011732	3200	3209#
TST145	012030	3232	3240#
TST146	012136	3264	3272#
TST147	012260	3302	3310#
TST15	004230	1658	1666#
TST150	012356	3329	3337#
TST151	012452	3359	3367#
TST152	012554	3386	3394#
TST153	012650	3416	3424#
TST154	012764	3453	3461#

FPU BASIC INSTR TESTS MACY11 27(1006)  
 DQFPAA.P11 09-FEB-77 09:42

09-FEB-77 09:48 PAGE 159  
 CROSS REFERENCE TABLE -- USER SYMBOLS

TST155	013072	3484	3492#
TST156	013154	3508	3516#
TST157	013254	3538	3546#
TST16	004304	1681	1689#
TST160	013344	3564	3577#
TST161	013674	3644#	
TST162	013726	3648	3661#
TST163	013760	3665	3678#
TST164	014012	3682	3695#
TST165	014044	3699	3712#
TST166	014076	3716	3729#
TST167	014130	3733	3746#
TST17	004400	1710	1718#
TST170	014162	3750	3763#
TST171	014214	3767	3780#
TST172	014246	3784	3797#
TST173	014300	3801	3814#
TST174	014332	3818	3832#
TST175	014374	3836	3851#
TST176	014436	3855	3870#
TST177	014500	3874	3889#
TST2	003370	1419#	
TST20	004466	1742	1750#
TST200	014542	3893	3908#
TST201	014604	3912	3927#
TST202	014646	3931	3946#
TST203	014710	3950	3965#
TST204	014752	3969	3984#
TST205	015014	3988	4003#
TST206	015056	4007	4022#
TST207	015120	4026	4042#
TST21	004530	1763	1771#
TST210	015152	4046	4059#
TST211	015204	4063	4076#
TST212	015236	4080	4093#
TST213	015270	4097	4110#
TST214	015322	4114	4127#
TST215	015354	4131	4144#
TST216	015406	4148	4161#
TST217	015440	4165	4178#
TST22	004570	1784	1792#
TST220	015472	4182	4195#
TST221	015524	4199	4212#
TST222	015556	4216	4230#
TST223	015620	4234	4249#
TST224	015662	4253	4268#
TST225	015724	4272	4287#
TST226	015766	4291	4306#
TST227	016030	4310	4325#
TST23	004646	1811	1824#
TST230	016072	4329	4344#
TST231	016134	4348	4363#
TST232	016176	4367	4382#
TST233	016240	4386	4401#
TST234	016302	4405	4420#
TST235	016344	4424	4440#

FPU BASIC INSTR TESTS MACY11 27(1006)  
 DDFPAA.P11 09-FEB-77 09:42

09-FEB-77 09:48 PAGE 160  
 CROSS REFERENCE TABLE -- USER SYMBOLS

TST236	016376	4444	4457#
TST237	016430	4461	4474#
TST24	005000	1845	1854#
TST240	016462	4478	4491#
TST241	016514	4495	4508#
TST242	016546	4512	4525#
TST243	016600	4529	4543#
TST244	016642	4547	4562#
TST245	016704	4566	4581#
TST246	016746	4585	4600#
TST247	017010	4604	4619#
TST25	005046	1859	1872#
TST250	017052	4623	4638#
TST251	017114	4642	4658#
TST252	017146	4662	4675#
TST253	017200	4679	4692#
TST254	017232	4696	4709#
TST255	017264	4713	4726#
TST256	017316	4727	4743#
TST257	017350	4747	4760#
TST26	005114	1877	1890#
TST260	017402	4764	4777#
TST261	017434	4781	4794#
TST262	017466	4798	4811#
TST263	017520	4815	4828#
TST264	017552	4832	4846#
TST265	017614	4850	4865#
TST266	017656	4869	4884#
TST267	017720	4888	4903#
TST27	005162	1895	1908#
TST270	017762	4907	4922#
TST271	020024	4926	4941#
TST272	020066	4945	4960#
TST273	020130	4964	4979#
TST274	020172	4983	4998#
TST275	020234	5002	5017#
TST276	020276	5021	5036#
TST277	020340	5040	5055#
TST3	003454	1437#	
TST30	005230	1913	1926#
TST31	005276	1931	1944#
TST32	005344	1949	1962#
TST33	005364	1967	1974#
TST34	005404	1979	1986#
TST35	005424	1991	1998#
TST36	005444	2003	2010#
TST37	005464	2015	2022#
TST4	003502	1453#	
TST40	005504	2027	2035#
TST41	005524	2040	2047#
TST42	005544	2052	2059#
TST43	005564	2064	2071#
TST44	005604	2076	2083#
TST45	005624	2088	2095#
TST46	005644	2100	2108#
TST47	005664	2113	2120#







2287#	2301#	2320#	2339#	2358#	2377#	2396#	2416#	2428#	2440#	2452#	2464#	2476#
2489#	2501#	2513#	2525#	2537#	2549#	2562#	2574#	2586#	2598#	2610#	2622#	2635#
2647#	2659#	2671#	2683#	2696#	2708#	2720#	2732#	2744#	2763#	2795#	2832#	2872#
2909#	2947#	2986#	3025#	3058#	3097#	3135#	3167#	3206#	3237#	3269#	3307#	3334#
3364#	3391#	3421#	3458#	3489#	3513#	3543#	3574#	3640#	3642#	3657#	3659#	3674#
3676#	3691#	3693#	3708#	3710#	3725#	3727#	3742#	3744#	3759#	3761#	3776#	3778#
3793#	3795#	3810#	3812#	3828#	3830#	3847#	3849#	3866#	3868#	3885#	3887#	3904#
3906#	3923#	3925#	3942#	3944#	3961#	3963#	3980#	3982#	3999#	4001#	4018#	4020#
4038#	4040#	4055#	4057#	4072#	4074#	4089#	4091#	4106#	4108#	4123#	4125#	4140#
4142#	4157#	4159#	4174#	4176#	4191#	4193#	4208#	4210#	4226#	4228#	4245#	4247#
4264#	4266#	4283#	4285#	4302#	4304#	4321#	4323#	4340#	4342#	4359#	4361#	4378#
4380#	4397#	4399#	4416#	4418#	4436#	4438#	4453#	4455#	4470#	4472#	4487#	4489#
4504#	4506#	4521#	4523#	4539#	4541#	4558#	4560#	4577#	4579#	4596#	4598#	4615#
4617#	4634#	4636#	4654#	4656#	4671#	4673#	4688#	4690#	4705#	4707#	4722#	4724#
4739#	4741#	4756#	4758#	4773#	4775#	4790#	4792#	4807#	4809#	4824#	4826#	4842#
4844#	4861#	4863#	4880#	4882#	4899#	4901#	4918#	4920#	4937#	4939#	4956#	4958#
4975#	4977#	4994#	4996#	5013#	5015#	5032#	5034#					
6026#	6055#	6068#										
6021#	6025#	6030#	6033#	6044#	6070#							
5682#	5698#	5706#	5716#	5725#								
1091#	1308#	1360#	5103#	5104#	5712#	5729#						
992#												
6145#	6152#											
6147#												
1284#	6112#	6142#										
6145#												
6122#	6128#											
1079#	5790#	5937#										
6108#												
6108#												
6108#												
6108#												
1034#												
1036#	1401#	1402#	1403#	1404#	1406#	1423#	1424#	1425#	1426#	1428#	1440#	1441#
1456#	1457#	1472#	1473#	1488#	1489#	1503#	1505#	1511#	1525#	1527#	1539#	1553#
1555#	1562#	1572#	1586#	1588#	1599#	1613#	1615#	1621#	1626#	1640#	1642#	1653#
1667#	1669#	1676#	1690#	1692#	1705#	1719#	1721#	1737#	1751#	1753#	1758#	1772#
1774#	1779#	1793#	1795#	1806#	1829#	1830#	1831#	1840#	2768#	2773#	2778#	2800#
2805#	2810#	2815#	2837#	2842#	2855#	2877#	2882#	2892#	2914#	2919#	2925#	2930#
2952#	2957#	2969#	2991#	2996#	3008#	3030#	3035#	3041#	3063#	3068#	3086#	3102#
3107#	3118#	3140#	3145#	3150#	3172#	3177#	3189#	3211#	3214#	3225#	3242#	3245#
3257#	3274#	3277#	3295#	3312#	3315#	3322#	3339#	3342#	3352#	3369#	3372#	3379#
3396#	3399#	3404#	3409#	3426#	3429#	3436#	3446#	3463#	3466#	3477#	3494#	3497#
3501#	3518#	3521#	3533#	3548#	3551#	3557#	3579#	3581#	3584#	3587#	3589#	3591#
3593#	3594#	5158#	5175#	5179#	5183#	5187#	5193#	5199#	5233#	5252#	5258#	5264#
5270#	5279#	5288#	5301#	5305#	5323#	5344#	5350#	5368#	5392#	5396#	5414#	5435#
5441#	5459#	5483#	5487#	5505#	5526#	5532#	5550#	5575#	5579#	5597#	5618#	5624#
5642#	6649#	6650#	6651#	6662#	6666#	6670#	6675#	6677#	6679#	6681#	6683#	6685#
6687#	6689#	6691#	6722#	6724#	6726#	6729#	6731#	6735#	6739#	6743#	6747#	
1037#	1405#	1406#	1427#	1428#	1508#	1511#	1530#	1539#	1558#	1572#	1592#	1599#
1619#	1626#	1646#	1653#	1672#	1676#	1703#	1705#	1729#	1732#	1737#	1755#	1758#
1776#	1779#	1798#	1801#	1806#	1838#	1839#	1840#	2769#	2780#	2801#	2817#	2838#
2857#	2878#	2894#	2915#	2932#	2953#	2971#	2992#	3010#	3031#	3043#	3064#	3103#
3120#	3141#	3152#	3173#	3191#	3212#	3227#	3243#	3259#	3275#	3297#	3313#	3324#
3340#	3354#	3370#	3381#	3397#	3411#	3427#	3448#	3464#	3479#	3495#	3503#	3519#
3549#	3559#	3580#	3583#	3586#	3588#	3590#	3592#	3595#	3599#	3602#	3606#	3609#

SOCNT 025264  
 SOMODE 025266  
 SOVER 023616  
 SPASS 001332  
 SPASTM 001006  
 SPOWER 025516  
 SPWRAD 025504  
 SPWRDN 025336  
 SPWRMG 025500  
 SPWRUP 025410  
 SQUES 001320  
 SRDCHR= \*\*\*\*\* U  
 SRDDEC= \*\*\*\*\* U  
 SRDLIN= \*\*\*\*\* U  
 SRDOCT= \*\*\*\*\* U  
 SREGAD 001166  
 SREGO 001170

SREG1 001172





2557	2562	2566#	2570	2574	2578#	2582	2586	2590#	2594	2598	2602#	2606
2610	2614#	2618	2622	2626#	2630	2635	2639#	2643	2647	2651#	2655	2659
2663#	2667	2671	2675#	2679	2683	2687#	2691	2696	2700#	2704	2708	2712#
2716	2720	2724#	2728	2732	2736#	2740	2744	2748#	2752	2763	2767#	2789
2795	2799#	2826	2832	2836#	2866	2872	2876#	2903	2909	2913#	2941	2947
2951#	2980	2986	2990#	3019	3025	3029#	3052	3058	3062#	3091	3097	3101#
3129	3135	3139#	3161	3167	3171#	3200	3206	3210#	3232	3237	3241#	3264
3269	3273#	3302	3307	3311#	3329	3334	3338#	3359	3364	3368#	3386	3391
3395#	3416	3421	3425#	3453	3458	3462#	3484	3489	3493#	3508	3513	3517#
3538	3543	3547#	3564	3574	3578#	3640	3645#	3648	3657	3662#	3665	3674
3679#	3682	3691	3696#	3699	3708	3713#	3716	3725	3730#	3733	3742	3747#
3750	3759	3764#	3767	3776	3781#	3784	3793	3798#	3801	3810	3815#	3818
3828	3833#	3836	3847	3852#	3855	3866	3871#	3874	3885	3890#	3893	3904
3909#	3912	3923	3928#	3931	3942	3947#	3950	3961	3966#	3969	3980	3985#
3988	3999	4004#	4007	4018	4023#	4026	4038	4043#	4046	4055	4060#	4063
4072	4077#	4080	4089	4094#	4097	4106	4111#	4114	4123	4128#	4131	4140
4145#	4148	4157	4162#	4165	4174	4179#	4182	4191	4196#	4199	4208	4213#
4216	4226	4231#	4234	4245	4250#	4253	4264	4269#	4272	4283	4288#	4291
4302	4307#	4310	4321	4326#	4329	4340	4345#	4348	4359	4364#	4367	4378
4383#	4386	4397	4402#	4405	4416	4421#	4424	4436	4441#	4444	4453	4458#
4461	4470	4475#	4478	4487	4492#	4495	4504	4509#	4512	4521	4526#	4529
4539	4544#	4547	4558	4563#	4566	4577	4582#	4585	4596	4601#	4604	4615
4620#	4623	4634	4639#	4642	4654	4659#	4662	4671	4676#	4679	4688	4693#
4696	4705	4710#	4713	4722	4727#	4730	4739	4744#	4747	4756	4761#	4764
4773	4778#	4781	4790	4795#	4798	4807	4812#	4815	4824	4829#	4832	4842
4847#	4850	4861	4866#	4869	4880	4885#	4888	4899	4904#	4907	4918	4923#
4926	4937	4942#	4945	4956	4961#	4964	4975	4980#	4983	4994	4999#	5002
5013	5018#	5021	5032	5037#	5040	5052						
1029#	5926#	5937										
1033#	5875	5937										
1028#	5924	5937										
1282	6079#											
6090#	6101											
6094#	6103#	6104#	6105#	6106#								
6084	6101#											
991#												
1005#	1393#	5101#	5697	5719#	5720	5725	5729	5755	5790			
6106												
6106												
5875#	5968	6094	6102									
5905	5912	5919	5924#	5925								
5765	5804#											
5930	5932	5935#										
6024#	6103											
6023	6026#	6105										
6019#	6104											
1093#												
993#												
1100#												
5684#												
6020#	6024#	6034	6069#									
5681	5765											
952#	956#	966	967#	969#	971#	972#	978	979#	981#	983#	1002#	1082
1275	1290	1291	5134	5209	5297	5340	5388	5431	5479	5522	5571	5614
5728	5729	5790	5857#	5937	5989#	6124	6150					
5941	5944											

STPB 001160  
 STPFLG 001165  
 STPS 001156  
 STRAP 025270  
 STRAP2 025312  
 STRP = 000005  
 STRPAD 025324  
 STSTM 001004  
 STSTM 001102  
 STYPB= \*\*\*\*\* U  
 STYPS= \*\*\*\*\* U  
 STYPE 024312  
 STYPEC 024524  
 STYPER 024076  
 STYPEX 024572  
 STYPOC 025066  
 STYPON 025102  
 STYPOS 025042  
 SUNIT 001336  
 SUNITH 001010  
 SUSMR 001350  
 SXTSTR 023366  
 SOFILL 025265  
 \$40CAT= \*\*\*\*\* U  
 = 033566  
 .  
 . \$ASTA= \*\*\*\*\* U

FPU BASIC INSTR TESTS MACY11 27(1006) 09-FEB-77 09:48 PAGE 167  
DQFPAA.P11 09-FEB-77 09:42 CROSS REFERENCE TABLE -- USER SYMBOLS

.SX = 001000 978# 983

CMPFLT	7648	5158	5233	5323	5368	5414	5459	5505	5550	5597	5642				
COMMEN	9048														
COMM00	7648														
COMM01	7648														
COMM02	7648														
COMM03	7648														
COMM04	7648														
COMM05	7648														
COMM06	7648														
COMM07	7648														
COMM1	7648	3676	3693	3744	3761	3830	3887	3906	3963	3982	4040	4074	4108	4142	4247
	4285	4323	4361	4380	4399	4455	4489	4523	4541	4598	4617	4656	4690	4724	4758
	4792	4863	4901	4939	4977	5015									
COMM10	7648														
COMM11	7648														
COMM12	7648														
COMM13	7648														
COMM14	7648														
COMM15	7648														
COMM16	7648														
COMM17	7648														
COMM2	7648	3812	4210	4826											
COMM20	7648														
COMM21	7648														
COMM22	7648														
COMM23	7648														
COMM24	7648														
COMM25	7648														
COMM26	7648														
COMM27	7648														
COMM3	7648	3642	3659	3710	3727	3778	3795	3849	3868	3925	3944	4001	4057	4091	4125
	4159	4176	4193	4228	4266	4304	4342	4438	4472	4506	4560	4579	4636	4673	4707
	4741	4775	4809	4844	4882	4920	4958	4996							
COMM30	7648														
COMM31	7648														
COMM32	7648														
COMM33	7648														
COMM34	7648														
COMM35	7648														
COMM36	7648														
COMM37	7648														
COMM4	7648	4020	4418	5034											
COMM40	7648														
COMM41	7648														
COMM42	7648														
COMM43	7648														
COMM44	7648														
COMM45	7648														
COMM46	7648														
COMM47	7648														
ENDCOM	9048														
ERRCMP	7648	1511	1534	1539	1562	1567	1572	1594	1599	1621	1626	1648	1653	1676	1705
	1732	1737	1758	1779	1801	1806	2810	2850	2887	2925	2964	3081	3086	3113	3184
	3220	3252	3285	3290	3347	3404	3436	3441	3533						
ERRLUR	7648	5789													
ERROR	7988	948	1408	1430	1443	1459	1475	1491	1513	1536	1541	1564	1569	1574	1596







	4444	4461	4478	4495	4512	4529	4547	4566	4585	4604	4623	4642	4662	4679	4696
	4713	4730	4747	4764	4781	4798	4815	4832	4850	4869	4888	4907	4926	4945	4964
	4983	5002	5021	5040											
SLASH	904#	1320	1347												
SPACE	904#														
STARS	904#	964	975	977	984	998	1082	1085	1265	1267	1350	1352	1367	1369	1394
	1396	1416	1418	1434	1436	1450	1452	1466	1468	1482	1484	1495	1497	1499	1501
	1521	1523	1549	1551	1582	1584	1609	1611	1636	1638	1663	1665	1686	1688	1715
	1717	1747	1749	1768	1770	1789	1791	1817	1819	1821	1823	1851	1853	1869	1871
	1887	1889	1905	1907	1923	1925	1941	1943	1959	1961	1971	1973	1983	1985	1995
	1997	2007	2009	2019	2021	2032	2034	2044	2046	2056	2058	2068	2070	2080	2082
	2092	2094	2105	2107	2117	2119	2129	2131	2141	2143	2153	2155	2165	2167	2178
	2180	2190	2192	2202	2204	2214	2216	2226	2228	2239	2241	2251	2253	2263	2265
	2275	2277	2287	2289	2301	2303	2320	2322	2339	2341	2358	2360	2377	2379	2396
	2398	2416	2418	2428	2430	2440	2442	2452	2454	2464	2466	2476	2478	2489	2491
	2501	2503	2513	2515	2525	2527	2537	2539	2549	2551	2562	2564	2574	2576	2586
	2588	2598	2600	2610	2612	2622	2624	2635	2637	2647	2649	2659	2661	2671	2673
	2683	2685	2696	2698	2708	2710	2720	2722	2732	2734	2744	2746	2759	2761	2763
	2765	2795	2797	2832	2834	2872	2874	2909	2911	2947	2949	2986	2988	3025	3027
	3058	3060	3097	3099	3135	3137	3167	3169	3206	3208	3237	3239	3269	3271	3307
	3309	3334	3336	3364	3366	3391	3393	3421	3423	3458	3460	3489	3491	3513	3515
	3543	3545	3570	3572	3574	3576	3640	3643	3657	3660	3674	3677	3691	3694	3708
	3711	3725	3728	3742	3745	3759	3762	3776	3779	3793	3796	3810	3813	3828	3831
	3847	3850	3866	3869	3885	3888	3904	3907	3923	3926	3942	3945	3961	3964	3980
	3983	3999	4002	4018	4021	4038	4041	4055	4058	4072	4075	4089	4092	4106	4109
	4123	4126	4140	4143	4157	4160	4174	4177	4191	4194	4208	4211	4226	4229	4245
	4248	4264	4267	4283	4286	4302	4305	4321	4324	4340	4343	4359	4362	4378	4381
	4397	4400	4416	4419	4436	4439	4453	4456	4470	4473	4487	4490	4504	4507	4521
	4524	4539	4542	4558	4561	4577	4580	4596	4599	4615	4618	4634	4637	4654	4657
	4671	4674	4688	4691	4705	4708	4722	4725	4739	4742	4756	4759	4773	4776	4790
	4793	4807	4810	4824	4827	4842	4845	4861	4864	4880	4883	4899	4902	4918	4921
	4937	4940	4956	4959	4975	4978	4994	4997	5013	5016	5032	5035	5052	5089	5332
	5423	5514	5606	5668	5731	5790	5860	5939	5996	6073	6110	6126			
STATUS	764#														
SWRSU	904#	1292#													
TADD01	764#	2763	2795	2832	2872	2909	2947	2986	3025	3058	3097	3135	3167		
TADD02	764#	2777	2809	2849	2886	2924	2963	3007	3040	3112	3149	3183			
TADD01	764#	3206	3237	3269	3307	3334	3364	3391	3421	3458	3489	3513	3543		
TADD02	764#	3219	3251	3284	3321	3346	3378	3403	3435	3476	3500	3556			
TADDR1	764#	1499	1521	1549	1582	1609	1636	1663	1686	1715	1747	1768	1789		
TADDR2	764#	1510	1533	1561	1593	1620	1647	1675	1704	1731	1757	1778	1800		
TRMTRP	6094#														
TYPBIN	904#														
TYPDEC	904#														
TYPNAM	904#														
TYPNUM	904#														
TYPOCS	904#														
TYPOCT	904#														
TYPTXT	904#														
UPCODE	764#	6133													
SSCHRE	996#	1036	1037	1038	1039	1040	1041	1042	1043	1044	1045	1046	1047	1048	1049
	1050	1051													
SSCHTH	996#	1052	1053	1054	1055	1056	1057	1058	1059	1060	1061	1062	1063	1064	1065
	1066	1067	1068	1069	1070	1071	1072	1073	1074	1075					
SSESCA	904#														
SSNEWT	904#	1394	1416	1434	1450	1466	1482	1499	1521	1549	1582	1609	1636	1663	1686

	1715	1747	1768	1789	1821	1851	1869	1887	1905	1923	1941	1959	1971	1983	1995
	2007	2019	2032	2044	2056	2068	2080	2092	2105	2117	2129	2141	2153	2165	2178
	2190	2202	2214	2226	2239	2251	2263	2275	2287	2301	2320	2339	2358	2377	2396
	2416	2428	2440	2452	2464	2476	2489	2501	2513	2525	2537	2549	2562	2574	2586
	2598	2610	2622	2635	2647	2659	2671	2683	2696	2708	2720	2732	2744	2763	2795
	2832	2872	2909	2947	2986	3025	3058	3097	3135	3167	3206	3237	3269	3307	3334
	3364	3391	3421	3458	3489	3513	3543	3574	3640	3657	3674	3691	3708	3725	3742
	3759	3776	3793	3810	3828	3847	3866	3885	3904	3923	3942	3961	3980	3999	4018
	4038	4055	4072	4089	4106	4123	4140	4157	4174	4191	4208	4226	4245	4264	4283
	4302	4321	4340	4359	4378	4397	4416	4436	4453	4470	4487	4504	4521	4539	4558
	4577	4596	4615	4634	4654	4671	4688	4705	4722	4739	4756	4773	4790	4807	4824
	4842	4861	4880	4899	4918	4937	4956	4975	4994	5013	5032				
SSSET	6094#	6103	6104	6105											
SSSETM	1308#														
SSSKIP	904#	1516	1544	1577	1604	1631	1658	1681	1710	1742	1763	1784	1811	1845	1859
	1877	1895	1913	1931	1949	1967	1979	1991	2003	2015	2027	2040	2052	2064	2076
	2088	2100	2113	2125	2137	2149	2161	2173	2186	2198	2210	2222	2234	2247	2259
	2271	2283	2295	2309	2328	2347	2366	2385	2404	2424	2436	2448	2460	2472	2484
	2497	2509	2521	2533	2545	2557	2570	2582	2594	2606	2618	2630	2643	2655	2667
	2679	2691	2704	2716	2728	2740	2752	2789	2826	2866	2903	2941	2980	3019	3052
	3091	3129	3161	3200	3232	3264	3302	3329	3359	3386	3416	3453	3484	3508	3538
	3564	3648	3665	3682	3699	3716	3733	3750	3767	3784	3801	3818	3836	3855	3874
	3893	3912	3931	3950	3969	3988	4007	4026	4046	4063	4080	4097	4114	4131	4148
	4165	4182	4199	4216	4234	4253	4272	4291	4310	4329	4348	4367	4386	4405	4424
	4444	4461	4478	4495	4512	4529	4547	4566	4585	4604	4623	4642	4662	4679	4696
	4713	4730	4747	4764	4781	4798	4815	4832	4850	4869	4888	4907	4926	4945	4964
	4983	5002	5021	5040											
.EQUAT	764#	794													
.HEADE	764#	765													
.SBPAS	764#	5052													
.SETUP	764#	1262													
.STPAS	764#	1320													
.SACT1	764#	962													
.SAPT8	764#	1083#													
.SAPTH	764#	973													
.SAPTY	764#	5937													
.SCATC	764#	950													
.SCHTA	764#	996													
.SEOP	764#	5089													
.SERRO	764#	5729													
.SPOWE	764#	6108													
.SSCOP	764#	5666													
.STRAP	764#	6071													
.STYER	764#	5790													
.STYPE	764#	5858													
.STYPO	764#	5994													

. ABS. 033566 000

ERRORS DETECTED: 0  
DEFAULT GLOBALS GENERATED: 0

DSKZ:DQFPAA, DSKZ:DQFPAA.SEG/SOL/CRF=DQFPAA.MEM, DQFPAA.MAC, DQFPAA.P11  
RUN-TIME: 34 33 2 SECONDS

B14

FPU BASIC INSTR TESTS MACY11 27(1006) 09-FEB-77 09:48 PAGE 174  
DQFPA.P11 09-FEB-77 09:42 CROSS REFERENCE TABLE -- MACRO NAMES

RUN-TIME RATIO: 472/71=6.6  
CORE USED: 33K (66 PAGES)