

# KD11K

FLOAT PT BASIC INST SET  
MD-11-DQFPA-B

EP-DQFPA-B-DL-A

JUN 1977

COPYRIGHT © 1977

**digital**

FICHE 1 OF 1

MADE IN USA

The microfiche card displays a grid of 100 frames, arranged in 10 rows and 10 columns. Each frame contains a small table or chart, likely representing floating-point basic instructions for the MD-11-DQFPA-B system. The data is too small to read clearly but appears to be organized in a structured format.

B01

HDR1DQFPABSEQ

00010000

770608

PDP10 411

I D E N T I F I C A T I O N

PRODUCT CODE:           MAINDEC-11-DQFPA-B-D  
PRODUCT NAME:           PDP-11/6X - FP11-E FLOATING POINT UNIT  
                          BASIC INSTRUCTION TESTS  
DATE :                   MAY, 1977  
MAINTAINER:             DIAGNOSTIC GROUP  
AUTHOR:                 DON NORTH  
REVISED BY:             DON 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.

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

## 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 0-34074(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-DGPPA-B

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	HALT 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

\*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.

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 WFP 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.

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	FUNCTION
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.



MAINDEC-11-DGFPA-B

## 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:

$$.1\#[\text{WORD1-BIT}\langle 06:00 \rangle]\#[\text{WORD2-BIT}\langle 15:00 \rangle]$$

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

$$.1\#[\text{WORD1-BIT}\langle 06:00 \rangle]\#[\text{WORD2-BIT}\langle 15:00 \rangle]  
\#[\text{WORD3-BIT}\langle 15:00 \rangle]\#[\text{WORD4-BIT}\langle 15:00 \rangle]$$

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

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

MODEL	AVERAGE EXECUTION TIME PER PASS	
	SHORTEST PASS	LONGEST PASS
PDP-11/6X MICROCODE	0:01	1:15
PDP-11/6X W/FP11-E	0:01	1:15

-----  
TIMES SPECIFIED AS (MINUTES):(SECONDS)

SHORTEST PASS ::= PASS=1, NO ITERATIONS, USING:  
SWR=(004003) FOR PDP-11/6X MICROCODE  
SWR=(004002) FOR PDP-11/6X W/FP11-E

LONGEST PASS ::= PASS>=2, 2000. ITERATIONS/TEST, USING:  
SWR=(000003) FOR PDP-11/6X MICROCODE  
SWR=(000002) FOR PDP-11/6X W/FP11-E

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 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 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

- ACO-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 FP11 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 (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 TYPEOUT 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 TYPEOUT FACILITY. THE SUBROUTINE WILL, GIVEN THE INDEXING BYTE FROM THE ERROR CALL INSTRUCTION, PICK UP THE CORRECT ERROR MESSAGE VECTOR FROM SEARTB (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

MAINDEC-11-DQFPA-B

NECESSARY SOFTWARE COMMUNICATION HOOKS ARE PRESENT.

10.3 XXDP COMPATIBILITY

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



14	OPERATIONAL SWITCH SETTINGS
32	BASIC DEFINITIONS
188	TRAP CATCHER
197	STARTING ADDRESS(ES)
200	ACT11 HOOKS
211	APT PARAMETER BLOCK
234	COMMON TAGS
323	APT MAILBOX-ETABLE
350	ERROR POINTER TABLE
461	PROGRAM DEFINED COMMON TAGS
502	START OF PASS ROUTINE
510	INITIALIZE THE COMMON TAGS
536	T1 TEST OF FPS REGISTER BY RIPPLING A 1
659	T2 TEST OF FPS REGISTER BY RIPPLING A 0
678	T3 TEST OF SETF INSTRUCTION
695	T4 TEST OF SETD INSTRUCTION
712	T5 TEST OF SETI INSTRUCTION
729	T6 TEST OF SETL INSTRUCTION
747	T7 TEST OF LOAD-SRC MODE-0, STORE-DST MODE-7(PC)
770	T10 TEST OF LOAD-SRC MODE-1, STORE-DST MODE-3
799	T11 TEST OF LOAD-SRC MODE-2, STORE-DST MODE-5
833	T12 TEST OF LOAD-SRC MODE-3, STORE-DST MODE-6(PC)
861	T13 TEST OF LOAD-SRC MODE-4, STORE-DST MODE-0
889	T14 TEST OF LOAD-SRC MODE-5, STORE-DST MODE-3(PC)
917	T15 TEST OF LOAD-SRC MODE-6, STORE-DST MODE-7
941	T16 TEST OF LOAD-SRC MODE-7, STORE-DST MODE-2(PC)
971	T17 TEST OF LOAD-SRC MODE-2(PC), STORE-DST MODE-4
1004	T20 TEST OF LOAD-SRC MODE-3(PC), STORE-DST MODE-6
1026	T21 TEST OF LOAD-SRC MODE-6(PC), STORE-DST MODE-1
1048	T22 TEST OF LOAD-SRC MODE-7(PC), STORE-DST MODE-2
1081	T23 TEST OF CFCC INSTRUCTION
1112	T24 TEST FP AC0, D MODE, TEST PATTERN LDAD-1
1131	T25 TEST FP AC0, D MODE, TEST PATTERN LDAD-2
1150	T26 TEST FP AC0, D MODE, TEST PATTERN LDAD-3
1169	T27 TEST FP AC0, D MODE, TEST PATTERN LDAD-4
1188	T30 TEST FP AC0, D MODE, TEST PATTERN LDAD-5
1207	T31 TEST FP AC0, D MODE, TEST PATTERN LDAD-6
1226	T32 TEST FP AC1, D MODE, TEST PATTERN LDAD-1
1239	T33 TEST FP AC1, D MODE, TEST PATTERN LDAD-2
1252	T34 TEST FP AC1, D MODE, TEST PATTERN LDAD-3
1265	T35 TEST FP AC1, D MODE, TEST PATTERN LDAD-4
1278	T36 TEST FP AC1, D MODE, TEST PATTERN LDAD-5
1291	T37 TEST FP AC1, D MODE, TEST PATTERN LDAD-6
1305	T40 TEST FP AC2, D MODE, TEST PATTERN LDAD-1
1318	T41 TEST FP AC2, D MODE, TEST PATTERN LDAD-2
1331	T42 TEST FP AC2, D MODE, TEST PATTERN LDAD-3
1344	T43 TEST FP AC2, D MODE, TEST PATTERN LDAD-4
1357	T44 TEST FP AC2, D MODE, TEST PATTERN LDAD-5
1370	T45 TEST FP AC2, D MODE, TEST PATTERN LDAD-6
1384	T46 TEST FP AC3, D MODE, TEST PATTERN LDAD-1
1397	T47 TEST FP AC3, D MODE, TEST PATTERN LDAD-2
1410	T50 TEST FP AC3, D MODE, TEST PATTERN LDAD-3
1423	T51 TEST FP AC3, D MODE, TEST PATTERN LDAD-4
1436	T52 TEST FP AC3, D MODE, TEST PATTERN LDAD-5
1449	T53 TEST FP AC3, D MODE, TEST PATTERN LDAD-6
1463	T54 TEST FP AC4, D MODE, TEST PATTERN LDAD-1

1476	T55	TEST FP AC4, D MODE, TEST PATTERN LOAD-2
1489	T56	TEST FP AC4, D D MODE, TEST PATTERN LOAD-3
1502	T57	TEST FP AC4, D D MODE, TEST PATTERN LOAD-4
1515	T60	TEST FP AC4, D D MODE, TEST PATTERN LOAD-5
1529	T61	TEST FP AC5, D D MODE, TEST PATTERN LOAD-1
1542	T62	TEST FP AC5, D D MODE, TEST PATTERN LOAD-2
1555	T63	TEST FP AC5, D D MODE, TEST PATTERN LOAD-3
1568	T64	TEST FP AC5, D D MODE, TEST PATTERN LOAD-4
1581	T65	TEST FP AC5, D D MODE, TEST PATTERN LOAD-5
1596	T66	TEST FP AC0, F R MODE, TEST PATTERN LDF-1
1616	T67	TEST FP AC0, F MODE, TEST PATTERN LDF-2
1636	T70	TEST FP AC0, F MODE, TEST PATTERN LDF-3
1656	T71	TEST FP AC0, F F MODE, TEST PATTERN LDF-4
1676	T72	TEST FP AC0, F F MODE, TEST PATTERN LDF-5
1696	T73	TEST FP AC0, F MODE, TEST PATTERN LDF-6
1717	T74	TEST FP AC1, F MODE, TEST PATTERN LDF-1
1730	T75	TEST FP AC1, F MODE, TEST PATTERN LDF-2
1743	T76	TEST FP AC1, F F MODE, TEST PATTERN LDF-3
1756	T77	TEST FP AC1, F F MODE, TEST PATTERN LDF-4
1769	T100	TEST FP AC1, F MODE, TEST PATTERN LDF-5
1782	T101	TEST FP AC1, F MODE, TEST PATTERN LDF-6
1796	T102	TEST FP AC2, F MODE, TEST PATTERN LDF-1
1809	T103	TEST FP AC2, F MODE, TEST PATTERN LDF-2
1822	T104	TEST FP AC2, F MODE, TEST PATTERN LDF-3
1835	T105	TEST FP AC2, F MODE, TEST PATTERN LDF-4
1848	T106	TEST FP AC2, F MODE, TEST PATTERN LDF-5
1861	T107	TEST FP AC2, F MODE, TEST PATTERN LDF-6
1875	T110	TEST FP AC3, F MODE, TEST PATTERN LDF-1
1888	T111	TEST FP AC3, F MODE, TEST PATTERN LDF-2
1901	T112	TEST FP AC3, F MODE, TEST PATTERN LDF-3
1914	T113	TEST FP AC3, F MODE, TEST PATTERN LDF-4
1927	T114	TEST FP AC3, F MODE, TEST PATTERN LDF-5
1940	T115	TEST FP AC3, F MODE, TEST PATTERN LDF-6
1954	T116	TEST FP AC4, F MODE, TEST PATTERN LDF-1
1967	T117	TEST FP AC4, F MODE, TEST PATTERN LDF-2
1980	T120	TEST FP AC4, F MODE, TEST PATTERN LDF-3
1993	T121	TEST FP AC4, F MODE, TEST PATTERN LDF-4
2006	T122	TEST FP AC4, F MODE, TEST PATTERN LDF-5
2020	T123	TEST FP AC5, F MODE, TEST PATTERN LDF-1
2033	T124	TEST FP AC5, F MODE, TEST PATTERN LDF-2
2046	T125	TEST FP AC5, F MODE, TEST PATTERN LDF-3
2059	T126	TEST FP AC5, F MODE, TEST PATTERN LDF-4
2072	T127	TEST FP AC5, F MODE, TEST PATTERN LDF-5
2092	T130	TEST OF LOAD-FSRC MODE-1, STORE-FDST MODE-6(PC), D MODE
2125	T131	TEST OF LOAD-FSRC MODE-4, STORE-FDST MODE-3(PC), D MODE
2163	T132	TEST OF LOAD-FSRC MODE-7, STORE-FDST MODE-3, D MODE
2204	T133	TEST OF LOAD-FSRC MODE-6(PC), STORE-FDST MODE-2, D MODE
2242	T134	TEST OF LOAD-FSRC MODE-2, STORE-FDST MODE-1, D MODE
2281	T135	TEST OF LOAD-FSRC MODE-5, STORE-FDST MODE-7(PC), D MODE
2321	T136	TEST OF LOAD-FSRC MODE-2(PC), STORE-FDST MODE-7, D MODE
2361	T137	TEST OF LOAD-FSRC MODE-7(PC), STORE-FDST MODE-6, D MODE
2395	T140	TEST OF LOAD-FSRC MODE-3, STORE-FDST MODE-2(PC), D MODE
2435	T141	TEST OF LOAD-FSRC MODE-6, STORE-FDST MODE-4, D MODE
2474	T142	TEST OF LOAD-FSRC MODE-3(PC), STORE-FDST MODE-0, D MODE
2507	T143	TEST OF LOAD-FSRC MODE-0, STORE-FDST MODE-5, D MODE
2547	T144	TEST OF LOAD-FSRC MODE-1, STORE-FDST MODE-4, F MODE

## TABLE OF CONTENTS

2579	T145	TEST OF LOAD-FSRC MODE-3, STORE-FDST MODE-6, F MODE
2612	T146	TEST OF LOAD-FSRC MODE-5, STORE-FDST MODE-3, F MODE
2651	T147	TEST OF LOAD-FSRC MODE-7, STORE-FDST MODE-1, F MODE
2679	T150	TEST OF LOAD-FSRC MODE-3(PC), STORE-FDST MODE-2, F MODE
2710	T151	TEST OF LOAD-FSRC MODE-7(PC), STORE-FDST MODE-7, F MODE
2738	T152	TEST OF LOAD-FSRC MODE-2, STORE-FDST MODE-3(PC), F MODE
2769	T153	TEST OF LOAD-FSRC MODE-4, STORE-FDST MODE-5, F MODE
2807	T154	TEST OF LOAD-FSRC MODE-2(PC), STORE-FDST MODE-7(PC), F MODE
2839	T155	TEST OF LOAD-FSRC MODE-6(PC), STORE-FDST MODE-6(PC), F MODE
2864	T156	TEST OF LOAD-FSRC MODE-0, STORE-FDST MODE-2(PC), F MODE
2895	T157	TEST OF LOAD-FSRC MODE-6, STORE-FDST MODE-0, F MODE
2927	T160	TEST ALL FP-ACCUM ARE THERE
2994	T161	TEST OF ABSF INSTR, DATA SET ABSF-1
3012	T162	TEST OF ABSF INSTR, DATA SET ABSF-2
3030	T163	TEST OF ABSF INSTR, DATA SET ABSF-3
3048	T164	TEST OF ABSF INSTR, DATA SET ABSF-4
3066	T165	TEST OF ABSF INSTR, DATA SET ABSF-5
3084	T166	TEST OF ABSF INSTR, DATA SET ABSF-6
3102	T167	TEST OF ABSF INSTR, DATA SET ABSF-7
3120	T170	TEST OF ABSF INSTR, DATA SET ABSF-10
3138	T171	TEST OF ABSF INSTR, DATA SET ABSF-11
3156	T172	TEST OF ABSF INSTR, DATA SET ABSF-12
3174	T173	TEST OF ABSF INSTR, DATA SET ABSF-13
3193	T174	TEST OF ABSD INSTR, DATA SET ABSD-1
3213	T175	TEST OF ABSD INSTR, DATA SET ABSD-2
3233	T176	TEST OF ABSD INSTR, DATA SET ABSD-3
3253	T177	TEST OF ABSD INSTR, DATA SET ABSD-4
3273	T200	TEST OF ABSD INSTR, DATA SET ABSD-5
3293	T201	TEST OF ABSD INSTR, DATA SET ABSD-6
3313	T202	TEST OF ABSD INSTR, DATA SET ABSD-7
3333	T203	TEST OF ABSD INSTR, DATA SET ABSD-10
3353	T204	TEST OF ABSD INSTR, DATA SET ABSD-11
3373	T205	TEST OF ABSD INSTR, DATA SET ABSD-12
3393	T206	TEST OF ABSD INSTR, DATA SET ABSD-13
3414	T207	TEST OF NEGF INSTR, DATA SET NEGF-1
3432	T210	TEST OF NEGF INSTR, DATA SET NEGF-2
3450	T211	TEST OF NEGF INSTR, DATA SET NEGF-3
3468	T212	TEST OF NEGF INSTR, DATA SET NEGF-4
3486	T213	TEST OF NEGF INSTR, DATA SET NEGF-5
3504	T214	TEST OF NEGF INSTR, DATA SET NEGF-6
3522	T215	TEST OF NEGF INSTR, DATA SET NEGF-7
3540	T216	TEST OF NEGF INSTR, DATA SET NEGF-10
3558	T217	TEST OF NEGF INSTR, DATA SET NEGF-11
3576	T220	TEST OF NEGF INSTR, DATA SET NEGF-12
3594	T221	TEST OF NEGF INSTR, DATA SET NEGF-13
3613	T222	TEST OF NEGD INSTR, DATA SET NEGD-1
3633	T223	TEST OF NEGD INSTR, DATA SET NEGD-2
3653	T224	TEST OF NEGD INSTR, DATA SET NEGD-3
3673	T225	TEST OF NEGD INSTR, DATA SET NEGD-4
3693	T226	TEST OF NEGD INSTR, DATA SET NEGD-5
3713	T227	TEST OF NEGD INSTR, DATA SET NEGD-6
3733	T230	TEST OF NEGD INSTR, DATA SET NEGD-7
3753	T231	TEST OF NEGD INSTR, DATA SET NEGD-10
3773	T232	TEST OF NEGD INSTR, DATA SET NEGD-11
3793	T233	TEST OF NEGD INSTR, DATA SET NEGD-12
3813	T234	TEST OF NEGD INSTR, DATA SET NEGD-13

## TABLE OF CONTENTS

3834	T235	TEST OF CLRF INSTR, DATA SET CLRF-1
3852	T236	TEST OF CLRF INSTR, DATA SET CLRF-2
3870	T237	TEST OF CLRF INSTR, DATA SET CLRF-3
3888	T240	TEST OF CLRF INSTR, DATA SET CLRF-4
3906	T241	TEST OF CLRF INSTR, DATA SET CLRF-5
3924	T242	TEST OF CLRF INSTR, DATA SET CLRF-6
3943	T243	TEST OF CLRD INSTR, DATA SET CLRD-1
3963	T244	TEST OF CLRD INSTR, DATA SET CLRD-2
3983	T245	TEST OF CLRD INSTR, DATA SET CLRD-3
4003	T246	TEST OF CLRD INSTR, DATA SET CLRD-4
4023	T247	TEST OF CLRD INSTR, DATA SET CLRD-5
4043	T250	TEST OF CLRD INSTR, DATA SET CLRD-6
4064	T251	TEST OF TSTF INSTR, DATA SET TSTF-1
4082	T252	TEST OF TSTF INSTR, DATA SET TSTF-2
4100	T253	TEST OF TSTF INSTR, DATA SET TSTF-3
4118	T254	TEST OF TSTF INSTR, DATA SET TSTF-4
4136	T255	TEST OF TSTF INSTR, DATA SET TSTF-5
4154	T256	TEST OF TSTF INSTR, DATA SET TSTF-6
4172	T257	TEST OF TSTF INSTR, DATA SET TSTF-7
4190	T260	TEST OF TSTF INSTR, DATA SET TSTF-10
4208	T261	TEST OF TSTF INSTR, DATA SET TSTF-11
4226	T262	TEST OF TSTF INSTR, DATA SET TSTF-12
4244	T263	TEST OF TSTF INSTR, DATA SET TSTF-13
4263	T264	TEST OF TSTD INSTR, DATA SET TSTD-1
4283	T265	TEST OF TSTD INSTR, DATA SET TSTD-2
4303	T266	TEST OF TSTD INSTR, DATA SET TSTD-3
4323	T267	TEST OF TSTD INSTR, DATA SET TSTD-4
4343	T270	TEST OF TSTD INSTR, DATA SET TSTD-5
4363	T271	TEST OF TSTD INSTR, DATA SET TSTD-6
4383	T272	TEST OF TSTD INSTR, DATA SET TSTD-7
4403	T273	TEST OF TSTD INSTR, DATA SET TSTD-10
4423	T274	TEST OF TSTD INSTR, DATA SET TSTD-11
4443	T275	TEST OF TSTD INSTR, DATA SET TSTD-12
4463	T276	TEST OF TSTD INSTR, DATA SET TSTD-13
4485		SUB PASS END CONTROL
4526		END OF PASS ROUTINE (MODIFIED SYSMAC)
4562		SUBR TO PERFORM TEST OF LDD/STD
4637		SUBR TO PERFORM TEST OF LDD/LDF/STD
4725		SUBR TO TEST THE ABSF INSTRUCTION
4768		SUBR TO TEST THE ABSD INSTRUCTION
4816		SUBR TO TEST THE NEGF INSTRUCTION
4859		SUBR TO TEST THE NEGD INSTRUCTION
4907		SUBR TO TEST THE CLRF INSTRUCTION
4950		SUBR TO TEST THE CLRD INSTRUCTION
4999		SUBR TO TEST THE TSTF INSTRUCTION
5042		SUBR TO TEST THE TSTD INSTRUCTION
5090		FPP TRAP CATCHER
5101		SCOPE HANDLER ROUTINE
5165		ERROR HANDLER ROUTINE
5228		ERROR MESSAGE TYPEOUT ROUTINE (MODIFIED SYSMAC)
5294		TYPE ROUTINE
5373		APT COMMUNICATIONS ROUTINE
5430		BINARY TO OCTAL (ASCII) AND TYPE
5507		TRAP DECODER
5530		TRAP TABLE
5544		POWER DOWN AND UP ROUTINES

FPU BASIC INSTR TESTS MACY11 27(1006) 25-APR-77 09:12  
DQFPAB.P11 19-APR-77 13:36 TABLE OF CONTENTS

H02

SEQ 0005

5591

ERROR MESSAGES, DATA HEADERS, DATA VECTORS, ETC

8 8

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

```
.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
	002000	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
	000002	1=TEST ONLY UNIT SPECIFIED IN SW<00>
0	000002	0=SELECT HFP, IF SW<01>=1
	000003	1=SELECT WFP, IF SW<01>=1

.SBTTL BASIC DEFINITIONS

```
.*INITIAL ADDRESS OF THE STACK POINTER *** 1100 ***
001100 STACK= 1100
.*EQUIV EMT,ERROR ;;BASIC DEFINITION OF ERROR CALL
.*EQUIV IOT,SCOPE ;;BASIC DEFINITION OF SCOPE CALL

.*MISCELLANEOUS DEFINITIONS
000011 HT= 11 ;;CODE FOR HORIZONTAL TAB
000012 LF= 12 ;;CODE FOR LINE FEED
000015 CR= 15 ;;CODE FOR CARRIAGE RETURN
000200 CRLF= 200 ;;CODE FOR CARRIAGE RETURN-LINE FEED
177776 PS= 177776 ;;PROCESSOR STATUS WORD
177774 .EQUIV PS,PSW
177774 STKLM= 177774 ;;STACK LIMIT REGISTER
177772 PIRQ= 177772 ;;PROGRAM INTERRUPT REQUEST REGISTER
177570 DSWR= 177570 ;;HARDWARE SWITCH REGISTER
177570 DDISP= 177570 ;;HARDWARE DISPLAY REGISTER

.*GENERAL PURPOSE REGISTER DEFINITIONS
000000 R0= %0 ;;GENERAL REGISTER
000001 R1= %1 ;;GENERAL REGISTER
000002 R2= %2 ;;GENERAL REGISTER
000003 R3= %3 ;;GENERAL REGISTER
000004 R4= %4 ;;GENERAL REGISTER
000005 R5= %5 ;;GENERAL REGISTER
```

57	000006	R6=	%6	::	GENERAL REGISTER
58	000007	R7=	%7	::	GENERAL REGISTER
59	000006	SP=	%6	::	STACK POINTER
60	000007	PC=	%7	::	PROGRAM COUNTER
61					
62		.*PRIORITY LEVEL DEFINITIONS			
63	000000	PR0=	0	::	PRIORITY LEVEL 0
64	000040	PR1=	40	::	PRIORITY LEVEL 1
65	000100	PR2=	100	::	PRIORITY LEVEL 2
66	000140	PR3=	140	::	PRIORITY LEVEL 3
67	000200	PR4=	200	::	PRIORITY LEVEL 4
68	000240	PR5=	240	::	PRIORITY LEVEL 5
69	000300	PR6=	300	::	PRIORITY LEVEL 6
70	000340	PR7=	340	::	PRIORITY LEVEL 7
71					
72		.*"SWITCH REGISTER" SWITCH DEFINITIONS			
73	100000	SW15=	100000		
74	040000	SW14=	40000		
75	020000	SW13=	20000		
76	010000	SW12=	10000		
77	004000	SW11=	4000		
78	002000	SW10=	2000		
79	001000	SW09=	1000		
80	000400	SW08=	400		
81	000200	SW07=	200		
82	000100	SW06=	100		
83	000040	SW05=	40		
84	000020	SW04=	20		
85	000010	SW03=	10		
86	000004	SW02=	4		
87	000002	SW01=	2		
88	000001	SW00=	1		
89		.EQUIV	SW09, SW9		
90		.EQUIV	SW08, SW8		
91		.EQUIV	SW07, SW7		
92		.EQUIV	SW06, SW6		
93		.EQUIV	SW05, SW5		
94		.EQUIV	SW04, SW4		
95		.EQUIV	SW03, SW3		
96		.EQUIV	SW02, SW2		
97		.EQUIV	SW01, SW1		
98		.EQUIV	SW00, SW0		
99					
100		.*DATA BIT DEFINITIONS (BIT00 TO BIT15)			
101	100000	BIT15=	100000		
102	040000	BIT14=	40000		
103	020000	BIT13=	20000		
104	010000	BIT12=	10000		
105	004000	BIT11=	4000		
106	002000	BIT10=	2000		
107	001000	BIT09=	1000		
108	000400	BIT08=	400		
109	000200	BIT07=	200		
110	000100	BIT06=	100		
111	000040	BIT05=	40		
112	000020	BIT04=	20		

113 000010  
114 000004  
115 000002  
116 000001  
117  
118  
119  
120  
121  
122  
123  
124  
125  
126  
127  
128  
129 000004  
130 000010  
131 000014  
132 000014  
133 000014  
134 000020  
135 000024  
136 000030  
137 000034  
138 000060  
139 000064  
140 000240  
141  
142  
143 076600  
144  
145 000022  
146  
147 000144  
148 000344  
149  
150  
151 000244  
152  
153  
154 000000  
155 000001  
156 000002  
157 000003  
158 000004  
159 000005  
160  
161  
162 052525  
163 125252  
164 007417  
165 170360  
166 177777  
167 100000  
168 077777

BIT03= 10  
BIT02= 4  
BIT01= 2  
BIT00= 1  
.EQUIV BIT09,BIT9  
.EQUIV BIT08,BIT8  
.EQUIV BIT07,BIT7  
.EQUIV BIT06,BIT6  
.EQUIV BIT05,BIT5  
.EQUIV BIT04,BIT4  
.EQUIV BIT03,BIT3  
.EQUIV BIT02,BIT2  
.EQUIV BIT01,BIT1  
.EQUIV BIT00,BIT0

;\*BASIC "CPU" TRAP VECTOR ADDRESSES  
ERRVEC= 4 ; TIME OUT AND OTHER ERRORS  
RESVEC= 10 ; RESERVED AND ILLEGAL INSTRUCTIONS  
TBITVEC= 14 ; "T" BIT  
TRTVEC= 14 ; TRACE TRAP  
BPTVEC= 14 ; BREAKPOINT TRAP (BPT)  
IOTVEC= 20 ; INPUT/OUTPUT TRAP (IOT) \*\*SCOPE\*\*  
PWRVEC= 24 ; POWER FAIL  
EMTVEC= 30 ; EMULATOR TRAP (EMT) \*\*ERROR\*\*  
TRAPVEC= 34 ; "TRAP" TRAP  
TKVEC= 60 ; TTY KEYBOARD VECTOR  
TPVEC= 64 ; TTY PRINTER VECTOR  
PIRQVEC= 240 ; PROGRAM INTERRUPT REQUEST VECTOR

;\*MED INSTR CODES  
MED= 076600 ; OPCODE  
RWHAMI= 022 ; READ WHAMI  
RFLAG= 144 ; READ FLAGS  
WFLAG= 344 ; WRITE FLAGS

;\*FLOATING POINT INTERRUPT VECTOR  
FPPVEC= 244

;\*FLOATING POINT REGISTER DEFINITIONS  
AC0= %0  
AC1= %1  
AC2= %2  
AC3= %3  
AC4= %4  
AC5= %5

;\*BIT PATTERNS FOR TESTS  
ALTP= 052525 ; 0101...01  
ALTN= 125252 ; 1010...10  
ALT4P= 007417 ; 0000111100001111  
ALT4N= 170360 ; 1111000011110000  
M1= 177777 ; 1111...11 MINUS ONE, ALL 1'S  
M0= 100000 ; 1000...00 MINUS ZERO  
LGP= 077777 ; 0111...11 LGST + NUM (1ST WD FLT)



169 177777  
170 000200  
171 100200  
172 000177  
173 100177  
174 040200  
175 140200  
176  
177  
178 147757  
179 000000  
180  
181  
182 177760  
183  
184  
185 104117  
186  
187  
188  
189 000000  
190  
191  
192  
193 000174  
194 000174 000000  
195 000176 000000  
196  
197 000200 000137 003000  
198  
199  
200  
201  
202  
203 000204  
204 000046  
205 000046 021016  
206 000052  
207 000052 000000  
208 000204  
209 001000  
210  
211  
212  
213  
214  
215 001000  
216 000024  
217 000024 000200  
218 000044  
219 000044 001000  
220 001000  
221  
222  
223  
224

LGN= 17,777 ; 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  
SENDAD ; 1)SET LOC.46 TO ADDRESS OF SENDAD IN .SEOP  
.=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.

M02

225 001000  
226 001000 000000  
227 001002 001324  
228 001004 000001  
229 001006 000001  
230 001010 000000  
231 001012 000014  
232

SAPTHD:  
SHIBTS: .WORD 0  
SMBADR: .WORD \$MAIL  
STSTM: .WORD 1  
SPSTM: .WORD 1  
SUNITH: .WORD 0  
SETEND-SMAIL/2 ;;LENGTH MAILBOX-ETABLE(WORDS)  
:: TWO HIGH BITS OF 18 BIT MAILBOX ADDR.  
:: ADDRESS OF APT MAILBOX (BITS 0-15)  
:: RUN TIM OF LONGEST TEST  
:: RUN TIME IN SECS. OF 1ST PASS ON 1 UNIT (QUICK VERIFY)  
:: ADDITIONAL RUN TIME (SECS) OF A PASS FOR EACH ADDITIONAL UNIT

.SBTTL COMMON TAGS

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

233  
234  
235  
236  
237  
238  
239  
240  
241  
242  
243  
244  
245  
246  
247  
248  
249  
250  
251  
252  
253  
254  
255  
256  
257  
258  
259  
260  
261  
262  
263  
264  
265  
266  
267  
268  
269  
270  
271  
272  
273  
274  
275  
276  
277  
278  
279  
280  
281  
282  
283  
284  
285  
286  
287  
288

001100 001100  
001100 000000  
001102 000000  
001104 000000  
001106 000000  
001110 000000  
001112 000000  
001114 000000  
001116 000000  
001120 000001  
001122 000000  
001124 000000  
001126 000000  
001130 000000  
001132 000000  
001134 000000  
001136 000000  
001140 000  
001141 000  
001142 000000  
001144 177570  
001146 177570  
001150 000000  
001152 177560  
001154 177562  
001156 177564  
001160 177566  
001162 000  
001163 002  
001164 012  
001165 000  
001166 000000  
001170 000000  
001172 000000  
001174 000000  
001176 000000  
001200 000000  
001202 000000  
001204 000000  
001206 000000  
001210 000000  
001212 000000  
001214 000000  
001216 000000  
001220 000000  
001222 000000

      .=1100  
SCMTAG:          ::START OF COMMON TAGS  
;-----START OF CLEAR COMMON TAGS-----  
      .WORD          0  
STSTNM: .WORD      00          ::CONTAINS THE TEST NUMBER  
SERFLG: .WORD      00          ::CONTAINS ERROR FLAG  
\$ICNT:  .WORD      00          ::CONTAINS SUBTEST ITERATION COUNT  
\$LPADR: .WORD      00          ::CONTAINS SCOPE LOOP ADDRESS  
\$LPERR: .WORD      00          ::CONTAINS SCOPE RETURN FOR ERRORS  
\$ERTTL: .WORD      00          ::CONTAINS TOTAL ERRORS DETECTED  
\$ITEMB: .WORD      00          ::CONTAINS ITEM CONTROL BYTE  
\$ERMAX: .WORD      1          ::CONTAINS MAX. ERRORS PER TEST  
\$ERRPC: .WORD      00          ::CONTAINS PC OF LAST ERROR INSTRUCTION  
\$GDADR: .WORD      00          ::CONTAINS ADDRESS OF 'GOOD' DATA  
\$BDADR: .WORD      00          ::CONTAINS ADDRESS OF 'BAD' DATA  
\$GDDAT: .WORD      00          ::CONTAINS 'GOOD' DATA  
\$BDDAT: .WORD      00          ::CONTAINS 'BAD' DATA  
      .WORD          0          ::RESERVED--NOT TO BE USED  
      .WORD          0  
\$AUTOB: .BYTE      00          ::AUTOMATIC MODE INDICATOR  
\$INTAG: .BYTE      00          ::INTERRUPT MODE INDICATOR  
;-----END OF CLEAR COMMON TAGS-----  
\$SWR:  .WORD      DSWR          ::ADDRESS OF SWITCH REGISTER  
\$DISPLA: .WORD      DDISP        ::ADDRESS OF DISPLAY REGISTER  
\$SLPTST: .WORD      0          ::CONTAINS TEST NUMBER TO LOOP UPON  
\$TKS:   177560          ::TTY KBD STATUS  
\$TKB:   177562          ::TTY KBD BUFFER  
\$STPS:  177564          ::TTY PRINTER STATUS REG. ADDRESS  
\$STPB:  177566          ::TTY PRINTER BUFFER REG. ADDRESS  
\$NULL:  .BYTE      0          ::CONTAINS NULL CHARACTER FOR FILLS  
\$FILLS: .BYTE      2          ::CONTAINS # OF FILLER CHARACTERS REQUIRED  
\$FILLC: .BYTE      12          ::INSERT FILL CHARS. AFTER A "LINE FEED"  
\$STPFLG: .BYTE      0          ::"TERMINAL AVAILABLE" FLAG (BIT(07)=0=YES)  
\$REGAD: .WORD      0          ::CONTAINS THE ADDRESS FROM WHICH (\$REGO) WAS OBTAINED  
\$REG0:  .WORD      0          ::CONTAINS ((\$REGAD)+0)  
\$REG1:  .WORD      0          ::CONTAINS ((\$REGAD)+2)  
\$REG2:  .WORD      0          ::CONTAINS ((\$REGAD)+4)  
\$REG3:  .WORD      0          ::CONTAINS ((\$REGAD)+6)  
\$REG4:  .WORD      0          ::CONTAINS ((\$REGAD)+10)  
\$REG5:  .WORD      0          ::CONTAINS ((\$REGAD)+12)  
\$REG6:  .WORD      0          ::CONTAINS ((\$REGAD)+14)  
\$REG7:  .WORD      0          ::CONTAINS ((\$REGAD)+16)  
\$REG10: .WORD      0          ::CONTAINS ((\$REGAD)+20)  
\$REG11: .WORD      0          ::CONTAINS ((\$REGAD)+22)  
\$REG12: .WORD      0          ::CONTAINS ((\$REGAD)+24)  
\$REG13: .WORD      0          ::CONTAINS ((\$REGAD)+26)  
\$REG14: .WORD      0          ::CONTAINS ((\$REGAD)+30)  
\$REG15: .WORD      0          ::CONTAINS ((\$REGAD)+32)

289 001224 000000  
 290 001226 000000  
 291 001230 000000  
 292 001232 000000  
 293 001234 000000  
 294 001236 000000  
 295 001240 000000  
 296 001242 000000  
 297 001244 000000  
 298 001246 000000  
 299 001250 000000  
 300 001252 000000  
 301 001254 000000  
 302 001256 000000  
 303 001260 000000  
 304 001262 000000  
 305 001264 000000  
 306 001266 000000  
 307 001270 000000  
 308 001272 000000  
 309 001274 000000  
 310 001276 000000  
 311 001300 000000  
 312 001302 000000  
 313 001304 000000  
 314 001306 000000  
 315 001310 000000  
 316 001312 000000  
 317 001314 177607  
 318 001320 077  
 319 001321 015  
 320 001322 000012  
 321  
 322  
 323  
 324  
 325  
 326 001324  
 327 001324 000000  
 328 001326 000000  
 329 001330 000000  
 330 001332 000000  
 331 001334 000000  
 332 001336 000000  
 333 001340 000000  
 334 001342 000000  
 335 001344  
 336 001344 000  
 337 001345 000  
 338 001346 000000  
 339 001350 000000  
 340 001352 000000  
 341  
 342  
 343  
 344

000377

```

$REG16: .WORD 0
$REG17: .WORD 0
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
SESCAPE: 0
SBELL: .ASCIZ <207><377><377>
SQUES: .ASCII /?/
SCRLF: .ASCII <15>
SLF: .ASCIZ <12>
*****
.SBTTL APT MAILBOX-ETABLE
*****
.EVEN
$MAIL:
$MSGTY: .WORD AMSGTY
$FATAL: .WORD AFATAL
$TESTN: .WORD ATESTN
$PASS: .WORD APASS
$DEVCT: .WORD ADEVCT
$UNIT: .WORD AUNIT
$MSGAD: .WORD AMSGAD
$MSGLG: .WORD AMSGLG
$ETABLE:
$ENV: .BYTE AENV
$ENVM: .BYTE AENVM
$$WREG: .WORD ASWREG
$USWR: .WORD AUSWR
$CPUOP: .WORD ACPUOP
*****
: APT MAILBOX
: MESSAGE TYPE CODE
: FATAL ERROR NUMBER
: TEST NUMBER
: PASS COUNT
: DEVICE COUNT
: I/O UNIT NUMBER
: MESSAGE ADDRESS
: MESSAGE LENGTH
: APT ENVIRONMENT TABLE
: ENVIRONMENT BYTE
: ENVIRONMENT MODE BITS
: APT SWITCH REGISTER
: USER SWITCHES
: CPU TYPE, OPTIONS
: BITS 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

```

```

: CONTAINS (($REGAD)+34)
: CONTAINS (($REGAD)+36)
: 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

```

C03

FPU BASIC INSTR TESTS MACY11 27(1006) 25-APR-77 09:12 PAGE 9  
DQFPAB.P11 19-APR-77 13:36 APT MAILBOX-ETABLE

SEQ 0013

345  
346  
347 001354  
348

;\*  
;\*  
\$ETEND:  
.MEXIT

BIT 9=FLOATING POINT PROCESSOR  
BIT 8=MEMORY MANAGEMENT

.SBTTL ERROR POINTER TABLE

SERRTB:

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

;\* EM ::POINTS TO THE ERROR MESSAGE  
;\* DH ::POINTS TO THE DATA HEADER  
;\* DT ::POINTS TO THE DATA  
;\* DF ::POINTS TO THE DATA FORMAT

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

\*\*\*\*\* SPECIFIC TEST VECTORS \*\*\*\*\*

368	001354	026331	030523	033126	EMV001:	.WORD	EMF,DHA,DTB	:	FPS RIPPLE 1
369	001362	026366	030523	033126	EMV002:	.WORD	EMG,DHA,DTB	:	FPS RIPPLE 0
370	001370	026423	030541	033122	EMV003:	.WORD	EMH,DHB,DTA	:	SETF
371	001376	026462	030541	033122	EMV004:	.WORD	EMI,DHB,DTA	:	SETD
372	001404	026517	030541	033122	EMV005:	.WORD	EMJ,DHB,DTA	:	SETI
373	001412	026556	030541	033122	EMV006:	.WORD	EMK,DHB,DTA	:	SETL
374					*****		SRC/DST ADDR MODES VECTORS	*****	
375	001420	026613	031273	033412	EMV007:	.WORD	EML,DHO,DTT	:	M05-M77 RESULT
376	001426	027005	031362	033440	EMV010:	.WORD	EMN,DHO,DTV	:	M13-M35 DSTREG
377	001434	026613	031362	033440	EMV011:	.WORD	EML,DHO,DTV	:	M13-M35 RESULT
378	001442	026703	031324	033424	EMV012:	.WORD	EMM,DHP,DTU	:	M21-M53 SRCREG
379	001450	027005	031324	033424	EMV013:	.WORD	EMN,DHP,DTU	:	M21-M53 DSTREG
380	001456	026613	031324	033424	EMV014:	.WORD	EML,DHP,DTU	:	M21-M53 RESULT, M63-M71 RESULT
381	001464	026703	031160	033354	EMV015:	.WORD	EMM,DHL,DTQ	:	M30-M67 SRCREG
382	001472	026613	031160	033354	EMV016:	.WORD	EML,DHL,DTQ	:	M30-M67 RESULT
383	001500	026703	031130	033342	EMV017:	.WORD	EMM,DHK,DTP	:	M42-M00 SRCREG
384	001506	026613	031130	033342	EMV020:	.WORD	EML,DHK,DTP	:	M42-M00 RESULT
385	001514	026703	031242	033400	EMV021:	.WORD	EMM,DHN,DTS	:	M54-M37 SRCREG
386	001522	026613	031242	033400	EMV022:	.WORD	EML,DHN,DTS	:	M54-M37 RESULT, M77-M24 RESULT
387	001530	026613	031211	033366	EMV023:	.WORD	EMM,DHM,DTR	:	M72-M27 RESULT
388	001536	027005	031105	033332	EMV024:	.WORD	EMN,DHJ,DTO	:	M27-M43 DSTREG
389	001544	026613	031105	033332	EMV025:	.WORD	EML,DHJ,DTO	:	M27-M43 RESULT
390	001552	026613	031062	033322	EMV026:	.WORD	EMM,DHI,DTN	:	M37-M62 RESULT, M67-M12 RESULT
391	001560	027005	031242	033400	EMV027:	.WORD	EMN,DHP,DTS	:	M77-M24 DSTREG

\*\*\*\*\* VECTOR FOR CFCC TEST \*\*\*\*\*

393	001566	027107	030523	033126	EMV030:	.WORD	EMO,DHA,DTB	:	CFCC INSTR
394					*****		FSRC/FDST D ADDR MODES VECTORS	*****	
395	001574	027165	031523	033502	EMV031:	.WORD	EMP,DHT,DTY	:	M15-M67 RESULT, M67-M25 RESULT
396	001602	027261	031457	033466	EMV032:	.WORD	EMQ,DHS,DTX	:	M44-M37 SRCREG
397	001610	027165	031457	033466	EMV033:	.WORD	EMP,DHS,DTX	:	M44-M37 RESULT
398	001616	027370	032174	033644	EMV034:	.WORD	EMR,DHAA,DTAF	:	M75-M34 DSTREG
399	001624	027165	032174	033644	EMV035:	.WORD	EMP,DHAA,DTAF	:	M75-M34 RESULT
400	001632	027370	031523	033502	EMV036:	.WORD	EMR,DHT,DTY	:	M67-M25 DSTREG
401	001640	027261	031567	033516	EMV037:	.WORD	EMQ,DHU,DTZ	:	M20-M13 SRCREG
402	001646	027165	031567	033516	EMV040:	.WORD	EMP,DHU,DTZ	:	M20-M13 RESULT
403	001654	027261	032112	033624	EMV041:	.WORD	EMQ,DHZ,DTAE	:	M51-M77 SRCREG
404	001662	027165	032112	033624	EMV042:	.WORD	EMP,DHZ,DTAE	:	M51-M77 RESULT

405	001670	027165	031711	033552	EMV043:	.WORD	EMP, DHM, DTAB	:	M27-M70	RESULT	
406	001676	027165	032037	033606	EMV044:	.WORD	EMP, DHY, DTAD	:	M77-M64	RESULT	
407	001704	027261	031764	033570	EMV045:	.WORD	EMQ, DHX, DTAC	:	M32-M27	SACREG	
408	001712	027165	031764	033570	EMV046:	.WORD	EMP, DHX, DTAC	:	M32-M27	RESULT, M00-M52	RESULT
409	001720	027370	031640	033534	EMV047:	.WORD	EMR, DHV, DTAA	:	M63-M41	DSTREG	
410	001726	027165	031640	033534	EMV050:	.WORD	EMP, DHV, DTAA	:	M63-M41	RESULT	
411	001734	027165	031420	033454	EMV051:	.WORD	EMP, DHR, DTH	:	M37-M03	RESULT	
412	001742	027370	031764	033570	EMV052:	.WORD	EMR, DHX, DTAC	:	M00-M52	DSTREG	
413							*****				
414	001750	027702	032371	033716	EMV053:	.WORD	FSRC/FDST F ADDR	:	MODES	VECTORS *****	
415	001756	027477	032371	033716	EMV054:	.WORD	EMU, DHAD, DTAI	:	M12-M45	DSTREG	
416	001764	027573	032505	033750	EMV055:	.WORD	EMS, DHAD, DTAI	:	M12-M45	RESULT	
417	001772	027477	032505	033750	EMV056:	.WORD	EMT, DHAF, DTAK	:	M34-M60	SACREG	
418	002000	027573	032776	034050	EMV057:	.WORD	EMS, DHAF, DTAK	:	M34-M60	RESULT	
419	002006	027702	032776	034050	EMV060:	.WORD	EMT, DHAJ, DTAO	:	M50-M32	SACREG	
420	002014	027477	032776	034050	EMV061:	.WORD	EMU, DHAJ, DTAO	:	M50-M32	DSTREG	
421	002022	027477	032563	033770	EMV062:	.WORD	EMS, DHAG, DTAL	:	M72-M11	RESULT	
422	002030	027702	032326	033702	EMV063:	.WORD	EMU, DHAC, DTAH	:	M37-M23	DSTREG	
423	002036	027477	032326	033702	EMV064:	.WORD	EMS, DHAC, DTAH	:	M37-M23	RESULT, M23-M37	RESULT
424	002044	027477	032717	034030	EMV065:	.WORD	EMS, DHAI, DTAN	:	M77-M75	RESULT	
425	002052	027573	032326	033702	EMV066:	.WORD	EMT, DHAC, DTAH	:	M23-M37	SACREG	
426	002060	027573	032641	034010	EMV067:	.WORD	EMT, DHAN, DTAM	:	M45-M54	SACREG	
427	002066	027702	032641	034010	EMV070:	.WORD	EMU, DHAN, DTAM	:	M45-M54	DSTREG	
428	002074	027477	032641	034010	EMV071:	.WORD	EMS, DHAN, DTAM	:	M45-M54	RESULT	
429	002102	027477	032441	033734	EMV072:	.WORD	EMS, DHAE, DTAJ	:	M27-M77	RESULT	
430	002110	027477	033062	033134	EMV073:	.WORD	EMS, DHAK, DTC	:	M67-M67	RESULT, M02-M27	RESULT
431	002116	027477	032263	033666	EMV074:	.WORD	EMS, DHAB, DTAG	:	M61-M01	RESULT	
432							*****				
433	002124	026260	030550	033134	EMV075:	.WORD	FP ACC ALL THERE ?	:	VECTOR	*****	
434							*****				
435	002132	026051	030606	033146	EMV076:	.WORD	EME, DHC, DTC	:	FP AC ALL THERE ?		
436	002140	026051	030606	033154	EMV077:	.WORD	FPS ERROR VECTORS	:	*****		
437	002146	026051	030606	033162	EMV100:	.WORD	EMA, DHD, DTD	:	FPS - ABS, NEG, CLR, TST F		
438							*****				
439	002154	026075	030622	033170	EMV101:	.WORD	EMA, DHD, DTE	:	FPS - LDD/STD, ABS, NEG, CLR, TST D		
440	002162	026075	030622	033202	EMV102:	.WORD	EMA, DHD, DTF	:	FPS - LDD/LDF/STD		
441	002170	026075	030622	033214	EMV103:	.WORD	FEC/FEA ERROR VECTORS	:	*****		
442							*****				
443	002176	030011	030720	033240	EMV104:	.WORD	EMB, DHE, DTG	:	FEC/FEA - ABS, NEG, CLR, TST F		
444	002204	030047	030720	033262	EMV105:	.WORD	EMB, DHE, DTH	:	FEC/FEA - LDD/STD, ABS, ..., CLR D		
445	002212	026125	030662	033226	EMV106:	.WORD	EMB, DHE, DTI	:	FEC/FEA - LDD/LDF/STD		
446	002220	026125	030720	033240	EMV107:	.WORD	*****				
447	002226	030111	030662	033226	EMV110:	.WORD	RESULT VECTORS	:	*****		
448	002234	030111	030720	033240	EMV111:	.WORD	EMV, DHG, DTK	:	RESULT - LDD/STD		
449	002242	030245	030662	033226	EMV112:	.WORD	EMW, DHG, DTL	:	RESULT - LDD/LDF/STD		
450	002250	030245	030720	033240	EMV113:	.WORD	EMC, DHF, DTJ	:	RESULT - ABSF		
451	002256	030157	030662	033226	EMV114:	.WORD	EMC, DHG, DTK	:	RESULT - ABSD		
452	002264	030157	030720	033240	EMV115:	.WORD	EMX, DHF, DTJ	:	RESULT - NEGF		
453							*****				
454	002272	026173	031016	033304	EMV116:	.WORD	EMX, DHG, DTK	:	RESULT - NEGD		
455							*****				
456	002300	030313	033120	034072	EMV117:	.WORD	EMZ, DHF, DTJ	:	RESULT - CLRF		
457	002306	030350	033120	034072	EMV120:	.WORD	EMZ, DHG, DTK	:	RESULT - CLRD		
458	002314	030421	033120	034072	EMV121:	.WORD	EMY, DHF, DTJ	:	RESULT - TSTF		
459	002322	030465	033120	034072	EMV122:	.WORD	EMY, DHG, DTK	:	RESULT - TSTD		
							*****				
							ILLEGAL TRAP CATCHER VECTOR	:	*****		
							EMD, DHH, DTM	:	ILLEGAL TRAP CATCHER		
							*****				
							PC MODE 2 WRONG INCRE VECTORS	:	*****		
							EMAA, DHAL, DTAP	:	+0		
							EMAB, DHAL, DTAP	:	+4		
							EMAC, DHAL, DTAP	:	+6		
							EMAD, DHAL, DTAP	:	+10		

```

460          .SBTTL PROGRAM DEFINED COMMON TAGS
461          :*VARIABLES
462 002330 000000 FPS: .WORD 0 ; FPS STORED HERE AFTER STFPS
463 002332 000000 FEC: .WORD 0 ; FEC STORED HERE AFTER STST
464 002334 000000 FEA: .WORD 0 ; FEA STORED HERE AFTER STST
465 002336 000000 FPPOPC: .WORD 0 ; OLD PC SAVED HERE AFTER TRAP
466 002340 000000 FPPOPS: .WORD 0 ; OLD PS SAVED HERE AFTER TRAP
467 002342 000000 FPPOSP: .WORD 0 ; SP AFTER TRAP
468 002344 000000 EXPFEA: .WORD 0 ; EXPECTED FEA
469
470          :*REGISTER CONTENTS, AT ERROR, STORED HERE
471 002346 000000 EREG0: .WORD 0
472 002350 000000 EREG1: .WORD 0
473 002352 000000 EREG2: .WORD 0
474 002354 000000 EREG3: .WORD 0
475 002356 000000 EREG4: .WORD 0
476 002360 000000 EREG5: .WORD 0
477 002362 000000 EREG6: .WORD 0
478 002364 000000 EREG7: .WORD 0
479
480          :*CONSTANTS
481 002366 052525 177777 125252 PREVAC: .WORD ALTP,M1,ALTN,0 ; PREV CONTENTS OF FLOAT AC
482 002374 000000
483
484
485          :*MESSAGES FOR BEGIN PROGRAM/START OF PASS/ETC
486 002376 005015 005012 042115 BGNMES: .ASCII <CR><LF><LF><LF>"MD-11-DQFPA-"
487 002404 030455 026461 050504
488 002412 050106 026501
489 002416 027102 .ASCII "B."
490 002420 027056 .ASCII ".:"
491 002422 042120 026520 030461 .ASCIZ "PDP-11/6X F.P.U. BASIC INSTRUCTION TESTS"<CR><LF>
492 002430 033057 020130 027106
493 002436 027120 027125 041040
494 002444 051501 041511 044440
495 002452 051516 051124 041525
496 002460 044524 047117 052040
497 002466 051505 051524 005015
498 002474 000
499 002475 015 050012 051501 NWPAS1: .ASCIZ <CR><LF>"PASS #"
500 002502 020123 000043
    
```



```

501 .SBTTL START OF PASS ROUTINE
502
503
504 ;;*****
505 .ENABL AMA ; BEGIN ASSEMBLING RELATIVE REFERENCES AS ABSOLUTE
506 ;;*****
507
508 003000 START:
509 .SBTTL INITIALIZE THE COMMON TAGS
510 ;;CLEAR THE COMMON TAGS ($CMTAG) AREA
511 003000 012706 001100 MOV $CMTAG,R6 ;:FIRST LOCATION TO BE CLEARED
512 003004 005026 CLR (R6)+ ;:CLEAR MEMORY LOCATION
513 003006 022706 001144 CMP $SWR,R6 ;:DONE?
514 003012 001374 BNE -6 ;:LOOP BACK IF NO
515 003014 012706 001100 MOV $STACK,SP ;:SETUP THE STACK POINTER
516 ;;INITIALIZE A FEW VECTORS
517 003020 012737 023664 000020 MOV $SCOPE,$IOTVEC ;:IOT VECTOR FOR SCOPE ROUTINE
518 003026 012737 000340 000022 MOV $340,$IOTVEC+2 ;:LEVEL 7
519 003034 012737 024142 000030 MOV $ERROR,$EMTVEC ;:EMT VECTOR FOR ERROR ROUTINE
520 003042 012737 000340 000032 MOV $340,$EMTVEC+2 ;:LEVEL 7
521 003050 012737 025576 000034 MOV $TRAP,$TRAPVEC ;:TRAP VECTOR FOR TRAP CALLS
522 003056 012737 000340 000036 MOV $340,$TRAPVEC+2 ;:LEVEL 7
523 003064 012737 025644 000024 MOV $SPWRDN,$PWRVEC ;:POWER FAILURE VECTOR
524 003072 012737 000340 000026 MOV $340,$PWRVEC+2 ;:LEVEL 7
525 003100 013737 020766 02076J MOV $ENDCT,$EOPCT ;:SETUP END-OF-PROGRAM COUNTER
526 003106 005037 001310 CLR $TIMES ;:INITIALIZE NUMBER OF ITERATIONS
527 003112 005037 001312 CLR $ESCAPE ;:CLEAR THE ESCAPE ON ERROR ADDRESS
528 003116 012737 000001 001120 MOV $1,$SERMAX ;:ALLOW ONE ERROR PER TEST
529 003124 012737 003124 00111J MOV $,$SLPADR ;:INITIALIZE THE LOOP ADDRESS FOR SCOPE
530 003132 012737 003132 001112 MOV $,$SLPERR ;:SETUP THE ERROR LOOP ADDRESS
531 ;;SIZE FOR A HARDWARE SWITCH REGISTER. IF NOT FOUND OR IT IS
532 ;;EQUAL TO A "-1" SETUP FOR A SOFTWARE SWITCH REGISTER.
533 003140 013746 000004 MOV $ERRVEC,-(SP) ;:SAVE ERROR VECTOR
534 003144 012737 003200 000004 MOV $64,$ERRVEC ;:SET UP ERROR VECTOR
535 003152 012737 177570 001144 MOV $DSWR,$SWR ;:SETUP FOR A HARDWARE SWICH REGISTER
536 003160 012737 177570 001146 MOV $DDISP,$DISPLAY ;:AND A HARDWARE DISPLAY REGISTER
537 003166 022777 177777 175750 CMP $-1,$SWR ;:TRY TO REFERENCE HARDWARE SWR
538 003174 001012 BNE 66$ ;:BRANCH IF NO TIMEOUT TRAP OCCURRED
539 ;:AND THE HARDWARE SWR IS NOT = -1
540 003176 000403 BR 65$ ;:BRANCH IF NO TIMEOUT
541 003200 012716 003206 64$: MOV $65$,(SP) ;:SET UP FOR TRAP RETURN
542 003204 000002 RTI
543 003206 012737 000176 001144 65$: MOV $SWREG,$SWR ;:POINT TO SOFTWARE SWR
544 003214 012737 000174 001146 MOV $DISPREG,$DISPLAY
545 003222 012637 000004 66$: MOV (SP)+,$ERRVEC ;:RESTORE ERROR VECTOR
546
547 003226 005037 001332 CLR $PASS ;:CLEAR PASS COUNT
548 003232 132737 000200 001345 BITB $APTSIZE,$ENVM ;:TEST USER SIZE UNDER APT
549 003240 001403 BEQ 67$ ;:YES,USE NON-APT SWITCH
550 003242 012737 001346 001144 MOV $SSWREG,$SWR ;:NO,USE APT SWITCH REGISTER
551 003250
552
553 ; SET UP FPP UNEXPECTED TRAP CATCHER - - - - -
554 003250 012737 023624 000244 MOV $FPPILT,$FPPVEC ; NEW PC AT FPP TRAP
555 003256 005037 000246 CLR $FPPVEC+2 ; NEW PS AT FPP TRAP
556

```

```

557 003262 104401 002376          TYPE      ,BGNMES          ; ID MESSAGE AT START
558
559 ;////////////////////////////////////
560 ; MESSAGE ON WHETHER OR NOT HFP UNIT IS PRESENT
561
562 003266 076600 000022          MED      ,RWHAMI          ;WHAMI INTO RO
563 003272 032700 000020          BIT      #BIT04,RO      ;IS THERE A HFP UNIT ?
564 003276 001403          BEQ      70$           ;NO, BR
565 003300 104401 003314          TYPE      ,68$         ;INDICATE FP11-E PRESENT
566 003304 000453          BR      NEWPAS        ;GO FOR SUBPASS INIT
567 003306 104401 003354 70$: TYPE      ,69$         ;INDICATE NO FP11-E
568 003312 000450          BR      NEWPAS        ;GO FOR SUBPASS INIT
569
570 003314 005015 020052 050106 68$: .ASCIZ <15><12>"* FP11-E HFP UNIT PRESENT *"<15><12>
571 003322 030461 042455 044040
572 003330 050106 052440 044516
573 003336 020124 051120 051505
574 003344 047105 020124 006452
575 003352 000012
576 003354 005015 020052 047516 69$: .ASCIZ <15><12>"* NO FP11-E HFP UNIT - ALL TESTS WFP ONLY *"<15><12>
577 003362 043040 030520 026461
578 003370 020105 043110 020120
579 003376 047125 052111 026440
580 003404 040440 046114 052040
581 003412 051505 051524 053440
582 003420 050106 047440 046116
583 003426 020131 006452 000012
584
585 .EVEN
586 ;////////////////////////////////////
587
588 ;*****
589 ;NEW PASS ENTERS HERE
590 ;*****
591
592
593 003434 012706 001100 NEWPAS: MOV      #STACK,SP          ;RESET STACK PTR
594
595 003440 032777 010000 175476 BIT      #BIT12,SWR          ;INHIBIT STATUS TYPEOUTS ?
596 003446 001011          BNE     SUBPAS          ;BR IF YES
597
598 003450 104401 002475          TYPE      ,NWPASi       ;"PASS #"
599 003454 013746 001332          MOV      $PASS,-(SP)    ;PASS COUNT INTO ...
600 003460 005216          INC      (SP)           ; 1-N RANGE
601 003462 104403          TYPOS          ;TYPE OCTAL
602 003464 006 000          .BYTE   6,0            ; 6 DIGITS, NO LEADING ZEROS
603 003466 104401 001321          TYPE      ,$CRLF        ;END THE LINE
604
605
606 ;*****
607 ;NEW SUBPASS ENTERS HERE
608 ;*****
609
610 003472 012706 001100 SUBPAS: MOV      #STACK,SP          ;RESET SP FOR INSURANCE
611
612 003476 076600 000022          MED      ,RWHAMI          ;GET WHAMI INTO RO

```



```

635
636
637
638 003576 000004
639 003600 012700 000001
640 003604 012737 003612 001112
641
642 003612 010037 001170
643 003616 042737 000020 001170
644 003624 170137 001170
645 003630 042737 030000 001170
646 003636 170237 001172
647 003642 023737 001170 001172
648 003650 001401
649 003652 104001
650 003654 000241
651 003656 006100
652 003660 103354
653
654
655
656
657
658
659
660 003662 000004
661 003664 012700 177776
662 003670 012737 003676 001112
663
664 003676 010037 001170
665 003702 042737 000020 001170
666 003710 170137 001170
667 003714 042737 030000 001170
668 003722 170237 001172
669 003726 023737 001170 001172
670 003734 001401
671 003736 104002
672 003740 000261
673 003742 006100
674 003744 103754

```

```

*****
; *TEST 1 TEST OF FPS REGISTER BY RIPPLING A 1
*****
TST1: SCOPE
MOV #BIT00,RO ; INITIAL PATTERN
MOV #1$,SLPERR ; ERROR LOOPING RETURN
1$: MOV RO,$REG0 ; EDIT PATTERN SO BIT 4 OF FPS ( FMM )
BIC #BIT04,$REG0 ; WILL ALWAYS BE OFF DURING TEST
LDFPS $REG0 ; LOAD FPS
BIC #BIT13!BIT12,$REG0 ; BITS 13 AND 12 ALWAYS READ 0
STFPS $REG1 ; STORE FPS
CMP $REG0,$REG1 ; LOAD/STORE WORK ?
BEQ 2$ ; YES
ERROR 1 ; NO - ERROR RETURN
2$: CLC ; ROTATE IN A 0
ROL RO ; SHIFT PATTERN 1 POSITION LEFT
BCB 1$ ; CONT IF NOT YET FINISHED

```

```

*****
; *TEST 2 TEST OF FPS REGISTER BY RIPPLING A 0
*****
TST2: SCOPE
MOV #1CBIT00,RO ; INITIAL PATTERN
MOV #1$,SLPERR ; ERROR LOOPING RETURN
1$: MOV RO,$REG0 ; EDIT PATTERN SO BIT 4 OF FPS ( FMM )
BIC #BIT04,$REG0 ; WILL ALWAYS BE OFF DURING TEST
LDFPS $REG0 ; LOAD FPS
BIC #BIT13!BIT12,$REG0 ; BITS 13 AND 12 ALWAYS READ 0
STFPS $REG1 ; STORE FPS
CMP $REG0,$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

```

675  
676  
677  
678 003746 000004  
679 003750 170127 147757  
680 003754 170001  
681 003756 170237 001170  
682 003762 022737 147557 001170  
683 003770 001401  
684 003772 104003  
685 003774

```

*****
: *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$:

```

686  
687  
688  
689  
690  
691  
692  
693  
694 003774 000004  
695 003776 170127 000000  
696 004002 170011  
697 004004 170237 001170  
698 004010 022737 000200 001170  
699 004016 001401  
700 004020 104004  
701 004022

```

*****
: *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$:

```

702  
703  
704  
705  
706  
707  
708  
709  
710 004022 000004  
711 004024 170127 147757  
712 004030 170002  
713 004032 170237 001170  
714 004036 022737 147657 001170  
715 004044 001401  
716 004046 104005  
717 004050

```

*****
: *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$:

```

718  
719  
720  
721  
722  
723  
724  
725  
726 004050 000004  
727 004052 170127 000000  
728 004056 170012  
729 004060 170237 001170  
730 004064 022737 000100 001170

```

*****
: *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 ?

```

L03

FPU BASIC INSTR TESTS MACY11 27(1006) 25-APR-77 09:12 PAGE 18  
DQFPAB.P11 19-APR-77 13:36 T6 TEST OF SETL INSTRUCTION

SEQ 0022

731 004072 001401  
732 004074 104006  
733 004076  
734

IS: BEQ 15  
ERROR 6

:  
: NO - ERROR  
: YES - NEXT TEST

```

735
736
737 ;:*****
738 .DSABL AMA ; ASSEMBLE ALL REFERENCES AS THEY ARE PRINTED
739 ;:*****
740
741 ;:*****
742 *TEST 7 TEST OF LOAD-SRC MODE-0, STORE-DST MODE-7(PC)
743 ;:*****
744 004076 000004 TST7: SCOPE
745 004100 016767 000036 175062 MOV ADDR1,$REG0 ; GET TEST PATTERN
746 004106 016705 175056 MOV $REG0,R5 ; DATA
747 004112 170105 LDFPS R5 ; M0-R5
748
749 004114 012767 001172 175052 MOV #$REG1,$REG2 ; ADDR(DEST)
750 004122 170277 175046 STFPS @REG2 ; M7-R7
751
752 004126 026767 175036 175036 CMP $REG0,$REG1 ; LOAD/STORE WORK?
753 004134 001401 BEQ 64$ ;
754 004136 104007 ERROR 7 ; NOT EQUAL, SIGNAL ERROR
755 004140 64$:
756
757 004140 000401 BR TST10 ;;
758
759 004142 105252 ADDR1: .WORD 105252 ; TEST PATTERN
760
761
762 ;:*****
763 *TEST 10 TEST OF LOAD-SRC MODE-1, STORE-DST MODE-3
764 ;:*****
765 004144 000004 TST10: SCOPE
766 004146 016767 000050 175014 MOV ADDR2,$REG0 ; GET TEST PATTERN
767
768 004154 012703 001170 MOV #$REG0,R3 ; ADDR(DATA)
769 004160 170113 LDFPS (R3) ; M1-R3
770
771 004162 012767 001172 175004 MOV #$REG1,$REG2 ; ADDR(DEST)
772 004170 012705 001174 MOV #$REG2,R5 ; ADDR(ADDR(DEST))
773 004174 170235 STFPS @R5+ ; M3-R5
774
775 004176 020527 001176 CMP R5,$REG2+2 ; WAS DST ADDR REG INCRE RIGHT AMOUNT?
776 004202 001401 BEQ 64$ ;
777 004204 104010 ERROR 10 ; NOT EQUAL, SIGNAL ERROR
778 004206 64$:
779
780 004206 026767 174756 174756 CMP $REG0,$REG1 ; LOAD/STORE WORK?
781 004214 001401 BEQ 65$ ;
782 004216 104011 ERROR 11 ; NOT EQUAL, SIGNAL ERROR
783 004220 65$:
784
785 004220 000401 BR TST11 ;;
786
787 004222 042505 ADDR2: .WORD 042505 ; TEST PATTERN
788
789
790 ;:*****

```

```

;#TEST 11 TEST OF LOAD-SRC MODE-2, STORE-DST MODE-5
;*****
TST11: SCOPE
MOV ADDR3,$REG0 ; GET TEST PATTERN
MOV #SREG0,R1 ; ADDR(DATA)
LDFPS (R1)+ ; M2-R1
MOV #SREG1,$REG2 ; ADDR(DEST)
MOV #SREG2+2,R3 ; ADDR(ADDR(DEST)+2)
STFPS @-(R3) ; M5-R3
CMP R1,#SREG0+2 ; WAS SRC ADDR REG INCRE RIGHT AMOUNT?
BEQ 64$ ;
ERROR 12 ; NOT EQUAL, SIGNAL ERROR
64$:
CMP R3,#SREG2 ; WAS DST ADDR REG DECRE RIGHT AMOUNT?
BEQ 65$ ;
ERROR 13 ; NOT EQUAL, SIGNAL ERROR
65$:
CMP $REG0,$REG1 ; LOAD/STORE WORK?
BEQ 66$ ;
ERROR 14 ; NOT EQUAL, SIGNAL ERROR
66$:
BR TST12 ;;
ADDR3: .WORD 105252 ; TEST PATTERN
    
```

```

;*****
;#TEST 12 TEST OF LOAD-SRC MODE-3, STORE-DST MODE-6(PC)
;*****
TST12: SCOPE
MOV ADDR4,$REG0 ; GET TEST PATTERN
MOV #SREG0,$REG2 ; ADDR(DATA)
MOV #SREG2,R0 ; ADDR(ADDR(DATA))
LDFPS @-(R0)+ ; M3-R0
STFPS $REG1 ; M6-R7
CMP R0,#SREG2+2 ; WAS SRC ADDR REG INCRE RIGHT AMOUNT?
BEQ 64$ ;
ERROR 15 ; NOT EQUAL, SIGNAL ERROR
64$:
CMP $REG0,$REG1 ; LOAD/STORE WORK?
BEQ 65$ ;
ERROR 16 ; NOT EQUAL, SIGNAL ERROR
65$:
BR TST13 ;;
    
```



847 004370 042505

ADDR4: .WORD 042505 ; TEST PATTERN

848  
849  
850  
851  
852  
853  
854  
855  
856  
857  
858  
859  
860  
861  
862  
863  
864  
865  
866  
867  
868  
869  
870  
871  
872  
873  
874  
875  
876  
877  
878  
879  
880  
881  
882  
883  
884  
885  
886  
887  
888  
889  
890  
891  
892  
893  
894  
895  
896  
897  
898  
899  
900  
901  
902

004372 000004  
004374 016767 000042 174566  
012702 001172  
170142  
170200  
010067 174554  
020227 001170  
001401  
104017  
026767 174536 174536  
001401  
104020  
000401  
105252  
000004  
016767 000046 174514  
012767 001170 174512  
012704 001176  
170154  
170237 001172  
020427 001174  
001401  
104021  
026767 174460 174460  
001401  
104022  
000401  
042505

\*\*\*\*\*  
\*TEST 13 TEST OF LOAD-SRC MODE-4, STORE-DST MODE-0  
\*\*\*\*\*

```
TST13: SCOPE
MOV ADDR5,$REG0 ; GET TEST PATTERN
MOV #SREG0+2,R2 ; ADDR(DATA+2)
LDFPS -(R2) ; M4-R2
STFPS R0 ; M0-R0
MOV R0,$REG1 ; DEST
CMP R2,#SREG0 ; 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 ;;
```

ADDR5: .WORD 105252 ; TEST PATTERN

\*\*\*\*\*  
\*TEST 14 TEST OF LOAD-SRC MODE-5, STORE-DST MODE-3(PC)  
\*\*\*\*\*

```
TST14: SCOPE
MOV ADDR6,$REG0 ; GET TEST PATTERN
MOV #SREG0,$REG2 ; ADDR(DATA)
MOV #SREG2+2,R4 ; ADDR(ADDR(DATA)+2)
LDFPS 2-(R4) ; M5-R4
STFPS 2#SREG1 ; M3-R7
CMP R4,#SREG2 ; 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 ;;
```

ADDR6: .WORD 042505 ; TEST PATTERN

903  
904  
905  
906  
907 004522 000004  
908 004524 016767 000044 174436  
909  
910 004532 012703 001254  
911 004536 170163 177714  
912  
913 004542 012767 001172 174424  
914 004550 012701 001134  
915 004554 170271 000040  
916  
917 004560 026767 174404 174404  
918 004566 001401  
919 004570 104014  
920 004572  
921  
922 004572 000401  
923  
924 004574 105252  
925  
926  
927  
928  
929  
930 004576 000004  
931 004600 016767 000064 174362  
932  
933 004606 012767 001170 174360  
934 004614 012702 001216  
935 004620 170172 177756  
936  
937 004624 012767 104117 000002  
938 004632 170227  
939 004634 000000  
940 004636 000403  
941 004640 104120  
942 004642 104121  
943 004644 104122  
944 004646 016767 177762 174316  
945  
946 004654 026767 174310 174310  
947 004662 001401  
948 004664 104023  
949 004666  
950  
951 004666 000401  
952  
953 004670 042505  
954  
955  
956  
957  
958

```

*****
; *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  40(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
*****

```

```

959 004672 000004 TST17: SCOPE
960 004674 016767 000056 174266 MOV ADDR11,$REG0 ; GET TEST PATTERN
961
962 004702 016767 174262 000002 MOV $REG0,15 ; PUT DATA
963 004710 170127 LDFPS (PC)+ ; M2-R7
964 004712 000000 1S: .WORD 0 ; DATA
965 004714 000403 BR 25 ; THIS SHOULD BE NEXT INSTR AFTER FPP PC MODE 2
966 004716 104120 ERROR 120 ; NOT HERE (+4 INCRE)
967 004720 104121 ERROR 121 ; OR HERE (+6 INCRE)
968 004722 104122 ERROR 122 ; OR HERE (+10 INCRE)
969
970 004724 012703 001174 2S: MOV #SREG1+2,R3 ; ADDR(DEST+2)
971 004730 170243 STFPS -(R3) ; M4-R3
972
973 004732 020327 001172 CMP R3,#SREG1 ; WAS DST ADDR REG DECRE RIGHT AMOUNT?
974 004736 001401 BEQ 64$ ;
975 004740 104024 ERROR 24 ; NOT EQUAL, SIGNAL ERROR
976 004742 64$:
977
978 004742 026767 174222 174222 CMP $REG0,$REG1 ; LOAD/STORE WORK?
979 004750 001401 BEQ 65$ ;
980 004752 104025 ERROR 25 ; NOT EQUAL, SIGNAL ERROR
981 004754 65$:
982
983 004754 000401 BR TST20 ;;
984
985 004756 104117 ADDR11: .WORD ERROR117 ; TEST PATTERN
986
987
988
989
990
991 004760 000004 TST20: SCOPE
992 004762 016767 000032 174200 MOV ADDR12,$REG0 ; GET TEST PATTERN
993
994 004770 170137 001170 LDFPS 2#$REG0 ; M3-R7
995
996 004774 012702 001144 MOV #SREG1-26,R2 ; ADDR(DEST)-26
997 005000 170262 000026 STFPS 26(R2) ; M6-R2
998
999 005004 026767 174160 174160 CMP $REG0,$REG1 ; LOAD/STORE WORK?
1000 005012 001401 BEQ 64$ ;
1001 005014 104026 ERROR 26 ; NOT EQUAL, SIGNAL ERROR
1002 005016 64$:
1003
1004 005016 000401 BR TST21 ;;
1005
1006 005020 042505 ADDR12: .WORD 042505 ; TEST PATTERN
1007
1008
1009
1010
1011
1012 005022 000004 TST21: SCOPE
1013 005024 016767 000030 174136 MOV ADDR13,$REG0 ; GET TEST PATTERN
1014

```

```

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

```

```

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

```

```

1015 005032 170167 174132 LDFPS $REG0 ; M6-R7
1016
1017 005036 012702 001172 MOV #SREG1,R2 ; ADDR(DEST)
1018 005042 170212 STFPS (R2) ; M1-R2
1019
1020 005044 026767 174120 174120 CMP $REG0,$REG1 ; LOAD/STORE WORK?
1021 005052 001401 BEQ 64$ ;
1022 005054 104026 ERROR 26 ; NOT EQUAL, SIGNAL ERROR
1023 005056 64$:
1024
1025 005056 000401 BR TST22 ;;
1026
1027 005060 105252 ADDR13: .WORD 105252 ; TEST PATTERN
1028
1029
1030

```

```

;*****
;*TEST 22 TEST OF LOAD-SRC MODE-7(PC), STORE-DST MODE-2
;*****

```

```

1031
1032
1033 005062 000004 TST22: SCOPE
1034 005064 016767 000046 174076 MOV ADDR14,$REG0 ; GET TEST PATTERN
1035
1036 005072 012767 001170 174074 MOV #SREG0,$REG2 ; ADDR(DATA)
1037 005100 170177 174070 LDFPS @SREG2 ; M7-R7
1038
1039 005104 012704 001172 MOV #SREG1,R4 ; ADDR(DEST)
1040 005110 170224 STFPS (R4)+ ; M2-R4
1041
1042 005112 020427 001174 CMP R4,#SREG1+2 ; WAS DST ADDR REG INCRE RIGHT AMOUNT?
1043 005116 001401 BEQ 64$ ;
1044 005120 104027 ERROR 27 ; NOT EQUAL, SIGNAL ERROR
1045 005122 64$:
1046
1047 005122 026767 174042 174042 CMP $REG0,$REG1 ; LOAD/STORE WORK?
1048 005130 001401 BEQ 65$ ;
1049 005132 104022 ERROR 22 ; NOT EQUAL, SIGNAL ERROR
1050 005134 65$:
1051
1052 005134 000401 BR TST23 ;;
1053
1054 005136 042505 ADDR14: .WORD 042505 ; TEST PATTERN
1055
1056

```

```

1057
1058
1059      ; ; *****
1060      ; ; .ENABL AMA ; BEGIN ASSEMBLING RELATIVE REFERENCES AS ABSOLUTE
1061      ; ; *****
1062      ; ; *****
1063      ; ; *TEST 23 TEST OF CFCC INSTRUCTION
1064      ; ; *****
1065      TST23: SCOPE
1066      MOV #10,R0 ; NUMBER OF ENTRIES IN TABLE
1067      MOV #0FCC,R1 ; START OF TABLE
1068      MOV #1$,SLPERR ; ERROR LOOPING RETURN
1069
1070      1$: MOV (R1),SREG0 ; GET FIRST FCC PATTERN
1071      BIC #CCONLY,SREG0 ; BITS IN FCC POSITIONS ONLY
1072      LDFPS SREG0 ; STORE IN FPS REGISTER
1073      BIT #BIT05,(R1) ; TEST WHETHER FLOATING A 1 OR 0
1074      BNE 2$
1075      CCC ; FOR FLOAT A 1, START W/ CC = 0000
1076      BR 3$
1077      2$: SCC ; FOR FLOAT A 0, START W/ CC = 1111
1078      3$: CFCC ; COPY THE CONDITION CODES
1079      MOV @#PSW,SREG1 ; GET CPU CC BITS
1080      BIC #CCONLY,SREG1 ; CLEAR EXTRANEIOUS BITS
1081      CMP SREG0,SREG1 ; WERE THEY COPIED OK ?
1082      BEQ 4$
1083      ERROR 30 ; NO - SIGNAL ERROR
1084      4$: ADD #2,R1 ; INCRE R1 OUT OF ERROR LOOP
1085      SOB R0,1$ ; LOOP CONTROL
1086      BR TST24
1087
1088      OFCC: .WORD 000001,000002,000004,000010 ; TABLE OF CC
1089      .WORD 000010
1090      .WORD 000056,000055,000053,000047 ; TEST PATTERNS
1091      .WORD 000047

```

```

1092
1093
1094
1095 005272 000004
1096 005274 012704 021214
1097 005300 012705 005312
1098 005304 004737 021032
1099
1100 005310 000413
1101
1102 005312
1103 005312 000000 000000 000000
1104 005320 000000
1105 005322 000000 000000 000000
1106 005330 000000
1107 005332 047653 047644
1108 005336 000000
1109
1110
1111
1112
1113 005340 000004
1114 005342 012704 021214
1115 005346 012705 005360
1116 005352 004737 021032
1117
1118 005356 000413
1119
1120 005360
1121 005360 177777 177777 177777
1122 005366 177777
1123 005370 177777 177777 177777
1124 005376 177777
1125 005400 047745 047750
1126 005404 000000
1127
1128
1129
1130
1131 005406 000004
1132 005410 012704 021214
1133 005414 012705 005426
1134 005420 004737 021032
1135
1136 005424 000413
1137
1138 005426
1139 005426 052525 052525 052525
1140 005434 052525
1141 005436 052525 052525 052525
1142 005444 052525
1143 005446 047652 047640
1144 005452 000000
1145
1146
1147

```

```

*****
; *TEST 24 TEST FP ACO, D MODE, TEST PATTERN LOAD-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 LOAD-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 LOAD-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 LOAD-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 LOAD-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 LOAD-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 LOAD-4
*****

```

```

1148
1149 005454 000004
1150 005456 012704 021214
1151 005462 012705 005474
1152 005466 004737 021032
1153
1154 005472 000413
1155
1156 005474
1157 005474 125252 125252 125252
1158 005502 125252
1159 005504 125252 125252 125252
1160 005512 125252
1161 005514 047704 047710
1162 005520 000000
1163
1164
1165
1166
1167 005522 000004
1168 005524 012704 021214
1169 005530 012705 005542
1170 005534 004737 021032
1171
1172 005540 000413
1173
1174 005542
1175 005542 100177 177777 177777
1176 005550 177777
1177 005552 100177 177777 177777
1178 005560 177777
1179 005562 043743 043754
1180 005566 000000
1181
1182
1183
1184
1185 005570 000004
1186 005572 012704 021214
1187 005576 012705 005610
1188 005602 004737 021032
1189
1190 005606 000413
1191
1192 005610
1193 005610 100000 000000 000000
1194 005616 000000
1195 005620 052525 177777 125252
1196 005626 000000
1197 005630 047602 147614
1198 005634 100014
1199
1200
1201
1202
1203 005636 000004

```

```

*****
†TST27: SCOPE
MOV #LDARDO,R4 ; PTR TO TESTING ROUTINE
MOV #LDAD4,R5 ; PTR TO TEST DATA SET
JSR PC,2#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,2#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,2#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

```

1204 005640 012704 021230  
1205 005644 012705 005312  
1206 005650 004737 021032

MOV #LDARD1,R4 ; PTR TO TESTING ROUTINE  
MOV #LDAD1,R5 ; PTR TO TEST DATA SET  
JSR PC,@#LDADT ; GO TEST

1207  
1208 005654 000400

BR TST33 ; ;

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

1209  
1210  
1211

\*\*\*\*\*  
; \*TEST 33 TEST FP AC1, D MODE, TEST PATTERN LDAD-2  
\*\*\*\*\*

1212  
1213  
1214

TST33: SCOPE  
MOV #LDARD1,R4 ; PTR TO TESTING ROUTINE  
MOV #LDAD2,R5 ; PTR TO TEST DATA SET  
JSR PC,@#LDADT ; GO TEST

1215 005656 000004  
1216 005660 012704 021230  
1217 005664 012705 005360  
1218 005670 004737 021032

1219  
1220 005674 000400

BR TST34 ; ;

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

1221  
1222  
1223

\*\*\*\*\*  
; \*TEST 34 TEST FP AC1, D MODE, TEST PATTERN LDAD-3  
\*\*\*\*\*

1224  
1225  
1226

TST34: SCOPE  
MOV #LDARD1,R4 ; PTR TO TESTING ROUTINE  
MOV #LDAD3,R5 ; PTR TO TEST DATA SET  
JSR PC,@#LDADT ; GO TEST

1227 005676 000004  
1228 005700 012704 021230  
1229 005704 012705 005426  
1230 005710 004737 021032

1231  
1232 005714 000400

BR TST35 ; ;

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

1233  
1234  
1235

\*\*\*\*\*  
; \*TEST 35 TEST FP AC1, D MODE, TEST PATTERN LDAD-4  
\*\*\*\*\*

1236  
1237  
1238

TST35: SCOPE  
MOV #LDARD1,R4 ; PTR TO TESTING ROUTINE  
MOV #LDAD4,R5 ; PTR TO TEST DATA SET  
JSR PC,@#LDADT ; GO TEST

1239 005716 000004  
1240 005720 012704 021230  
1241 005724 012705 005474  
1242 005730 004737 021032

1243  
1244 005734 000400

BR TST36 ; ;

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

1245  
1246  
1247

\*\*\*\*\*  
; \*TEST 36 TEST FP AC1, D MODE, TEST PATTERN LDAD-5  
\*\*\*\*\*

1248  
1249  
1250

TST36: SCOPE  
MOV #LDARD1,R4 ; PTR TO TESTING ROUTINE  
MOV #LDAD5,R5 ; PTR TO TEST DATA SET  
JSR PC,@#LDADT ; GO TEST

1251 005736 000004  
1252 005740 012704 021230  
1253 005744 012705 005542  
1254 005750 004737 021032

1255  
1256 005754 000400

BR TST37 ; ;

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

1257  
1258  
1259



```

1260
1261
1262
1263 005756 000004
1264 005760 012704 021230
1265 005764 012705 005610
1266 005770 004737 021032
1267
1268 005774 000400
1269
1270
1271
1272
1273
1274
1275
1276 005776 000004
1277 006000 012704 021244
1278 006004 012705 005312
1279 006010 004737 021032
1280
1281 006014 000400
1282
1283
1284
1285
1286
1287
1288 006016 000004
1289 006020 012704 021244
1290 006024 012705 005360
1291 006030 004737 021032
1292
1293 006034 000400
1294
1295
1296
1297
1298
1299
1300 006036 000004
1301 006040 012704 021244
1302 006044 012705 005426
1303 006050 004737 021032
1304
1305 006054 000400
1306
1307
1308
1309
1310
1311
1312 006056 000004
1313 006060 012704 021244
1314 006064 012705 005474
1315 006070 004737 021032

```

```

*****
;*TEST 37 TEST FP AC1, D MODE, TEST PATTERN LDAD-6
*****
†ST37: SCOPE
      MOV #LDAD1,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 #LDAD2,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 #LDAD2,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 #LDAD2,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 #LDAD2,R4 ; PTR TO TESTING ROUTINE
      MOV #LDAD4,R5 ; PTR TO TEST DATA SET
      JSR PC,@#LDADT ; GO TEST

```

K04

FPU BASIC INSTR TESTS MACY11 27(1006) 25-APR-77 09:12 PAGE 30  
DQFPAB.P11 19-APR-77 13:36

T43 TEST FP AC2, D MODE, TEST PATTERN LDAD-4

SEQ 0034

1316  
1317 006074 000400  
1318  
1319  
1320  
1321  
1322  
1323  
1324 006076 000004  
1325 006100 012704 021244  
1326 006104 012705 005542  
1327 006110 004737 021032  
1328  
1329 006114 000400  
1330  
1331  
1332  
1333  
1334  
1335  
1336 006116 000004  
1337 006120 012704 021244  
1338 006124 012705 005610  
1339 006130 004737 021032  
1340  
1341 006134 000400  
1342  
1343  
1344  
1345  
1346  
1347  
1348  
1349 006136 000004  
1350 006140 012704 021260  
1351 006144 012705 005312  
1352 006150 004737 021032  
1353  
1354 006154 000400  
1355  
1356  
1357  
1358  
1359  
1360  
1361 006156 000004  
1362 006160 012704 021260  
1363 006164 012705 005360  
1364 006170 004737 021032  
1365  
1366 006174 000400  
1367  
1368  
1369  
1370  
1371

```

BR TST44 ;;
; ( FOR DATA SEE TEST OF FP REG AC0 WITH TEST PATTERN LDAD-4, ABOVE )
;*****
; *TEST 44 TEST FP AC2, D MODE, TEST PATTERN LDAD-5
;*****
†ST44: SCOPE
MOV #LDARD2,R4 ; PTR TO TESTING ROUTINE
MOV #LDAD5,R5 ; PTR TO TEST DATA SET
JSR PC,@#LDADT ; GO TEST
BR TST45 ;;
; ( FOR DATA SEE TEST OF FP REG AC0 WITH TEST PATTERN LDAD-5, ABOVE )
;*****
; *TEST 45 TEST FP AC2, D MODE, TEST PATTERN LDAD-6
;*****
†ST45: SCOPE
MOV #LDARD2,R4 ; PTR TO TESTING ROUTINE
MOV #LDAD6,R5 ; PTR TO TEST DATA SET
JSR PC,@#LDADT ; GO TEST
BR TST46 ;;
; ( FOR DATA SEE TEST OF FP REG AC0 WITH TEST PATTERN LDAD-6, ABOVE )
;*****
; *TEST 46 TEST FP AC3, D MODE, TEST PATTERN LDAD-1
;*****
†ST46: SCOPE
MOV #LDARD3,R4 ; PTR TO TESTING ROUTINE
MOV #LDAD1,R5 ; PTR TO TEST DATA SET
JSR PC,@#LDADT ; GO TEST
BR TST47 ;;
; ( FOR DATA SEE TEST OF FP REG AC0 WITH TEST PATTERN LDAD-1, ABOVE )
;*****
; *TEST 47 TEST FP AC3, D MODE, TEST PATTERN LDAD-2
;*****
†ST47: SCOPE
MOV #LDARD3,R4 ; PTR TO TESTING ROUTINE
MOV #LDAD2,R5 ; PTR TO TEST DATA SET
JSR PC,@#LDADT ; GO TEST
BR TST50 ;;
; ( FOR DATA SEE TEST OF FP REG AC0 WITH TEST PATTERN LDAD-2, ABOVE )
;*****
; *TEST 50 TEST FP AC3, D MODE, TEST PATTERN LDAD-3

```

```

1372
1373 006176 000004
1374 006200 012704 021260
1375 006204 012705 005426
1376 006210 004737 021032
1377
1378 006214 000400
1379
1380
1381
1382
1383
1384
1385 006216 000004
1386 006220 012704 021260
1387 006224 012705 005474
1388 006230 004737 021032
1389
1390 006234 000400
1391
1392
1393
1394
1395
1396
1397 006236 000004
1398 006240 012704 021260
1399 006244 012705 005542
1400 006250 004737 021032
1401
1402 006254 000400
1403
1404
1405
1406
1407
1408
1409 006256 000004
1410 006260 012704 021260
1411 006264 012705 005610
1412 006270 004737 021032
1413
1414 006274 000400
1415
1416
1417
1418
1419
1420
1421
1422 006276 000004
1423 006300 012704 021274
1424 006304 012705 005312
1425 006310 004737 021032
1426
1427 006314 000400

```

```

*****
†T50: 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 AC0 WITH TEST PATTERN LDAD-3, ABOVE )
*****
; *TEST 51      TEST FP AC3, D MODE, TEST PATTERN LDAD-4
*****
†T51: 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 AC0 WITH TEST PATTERN LDAD-4, ABOVE )
*****
; *TEST 52      TEST FP AC3, D MODE, TEST PATTERN LDAD-5
*****
†T52: 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 AC0 WITH TEST PATTERN LDAD-5, ABOVE )
*****
; *TEST 53      TEST FP AC3, D MODE, TEST PATTERN LDAD-6
*****
†T53: 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 AC0 WITH TEST PATTERN LDAD-6, ABOVE )
*****
; *TEST 54      TEST FP AC4, D MODE, TEST PATTERN LDAD-1
*****
†T54: SCOPE
      MOV      #LDARD4,R4      ; PTR TO TESTING ROUTINE
      MOV      #LDAD1,R5      ; PTR TO TEST DATA SET
      JSR      PC,@#LDADT     ; GO TEST
      BR       TST55          ;;

```

1428  
1429  
1430  
1431  
1432  
1433  
1434 006316 000004  
1435 006320 012704 021274  
1436 006324 012705 005360  
1437 006330 004737 021032  
1438  
1439 006334 000400  
1440  
1441  
1442  
1443  
1444  
1445  
1446 006336 000004  
1447 006340 012704 021274  
1448 006344 012705 005426  
1449 006350 004737 021032  
1450  
1451 006354 000400  
1452  
1453  
1454  
1455  
1456  
1457  
1458 006356 000004  
1459 006360 012704 021274  
1460 006364 012705 005474  
1461 006370 004737 021032  
1462  
1463 006374 000400  
1464  
1465  
1466  
1467  
1468  
1469  
1470 006376 000004  
1471 006400 012704 021274  
1472 006404 012705 005542  
1473 006410 004737 021032  
1474  
1475 006414 000400  
1476  
1477  
1478  
1479  
1480  
1481  
1482  
1483 006416 000004

; ( FOR DATA SEE TEST OF FP REG AC0 WITH TEST PATTERN LDAD-1, ABOVE )

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

TST55: SCOPE  
MOV #LDAD4,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 AC0 WITH TEST PATTERN LDAD-2, ABOVE )

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

TST56: SCOPE  
MOV #LDAD4,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 AC0 WITH TEST PATTERN LDAD-3, ABOVE )

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

TST57: SCOPE  
MOV #LDAD4,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 AC0 WITH TEST PATTERN LDAD-4, ABOVE )

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

TST60: SCOPE  
MOV #LDAD4,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 AC0 WITH TEST PATTERN LDAD-5, ABOVE )

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

TST61: SCOPE

1484 006420 012704 021314  
 1485 006424 012705 005312  
 1486 006430 004737 021032  
 1487  
 1488 006434 000400  
 1489  
 1490  
 1491  
 1492  
 1493  
 1494  
 1495 006436 000004  
 1496 006440 012704 021314  
 1497 006444 012705 005360  
 1498 006450 004737 021032  
 1499  
 1500 006454 000400  
 1501  
 1502  
 1503  
 1504  
 1505  
 1506  
 1507 006456 000004  
 1508 006460 012704 021314  
 1509 006464 012705 005426  
 1510 006470 004737 021032  
 1511  
 1512 006474 000400  
 1513  
 1514  
 1515  
 1516  
 1517  
 1518  
 1519 006476 000004  
 1520 006500 012704 021314  
 1521 006504 012705 005474  
 1522 006510 004737 021032  
 1523  
 1524 006514 000400  
 1525  
 1526  
 1527  
 1528  
 1529  
 1530  
 1531 006516 000004  
 1532 006520 012704 021314  
 1533 006524 012705 005542  
 1534 006530 004737 021032  
 1535  
 1536 006534 000400  
 1537  
 1538  
 1539

```

MOV #LDARD5,R4 ; PTR TO TESTING ROUTINE
MOV #LDAD1,R5 ; PTR TO TEST DATA SET
JSR PC,@#LDADT ; GO TEST

BR TST62 ;;

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

;*****
;TEST 62 TEST FP ACS, D MODE, TEST PATTERN LOAD-2
;*****
TST62: SCOPE
MOV #LDARD5,R4 ; PTR TO TESTING ROUTINE
MOV #LDAD2,R5 ; PTR TO TEST DATA SET
JSR PC,@#LDADT ; GO TEST

BR TST63 ;;

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

;*****
;TEST 63 TEST FP ACS, D MODE, TEST PATTERN LOAD-3
;*****
TST63: SCOPE
MOV #LDARD5,R4 ; PTR TO TESTING ROUTINE
MOV #LDAD3,R5 ; PTR TO TEST DATA SET
JSR PC,@#LDADT ; GO TEST

BR TST64 ;;

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

;*****
;TEST 64 TEST FP ACS, D MODE, TEST PATTERN LOAD-4
;*****
TST64: SCOPE
MOV #LDARD5,R4 ; PTR TO TESTING ROUTINE
MOV #LDAD4,R5 ; PTR TO TEST DATA SET
JSR PC,@#LDADT ; GO TEST

BR TST65 ;;

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

;*****
;TEST 65 TEST FP ACS, D MODE, TEST PATTERN LOAD-5
;*****
TST65: SCOPE
MOV #LDARD5,R4 ; PTR TO TESTING ROUTINE
MOV #LDAD5,R5 ; PTR TO TEST DATA SET
JSR PC,@#LDADT ; GO TEST

BR TST66 ;;

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

B05

FPU BASIC INSTR TESTS MACY11 27(1006) 25-APR-77 09:12 PAGE 34  
DOFPAB.P11 19-APR-77 13:36 T65 TEST FP ACS, D MODE, TEST PATTERN LOAD-5

SEQ 0038

1540  
1541

```

1542
1543
1544
1545 006536 000004
1546 006540 012704 021514
1547 006544 012705 006556
1548 006550 004737 021334
1549
1550 006554 000415
1551
1552 006556
1553 006556 000000 000000 000000
1554 006564 000000
1555 006566 177777 177777
1556 006572 177777 177777 000000
1557 006600 000000
1558 006602 047547 047750
1559 006606 000000
1560
1561
1562
1563
1564 006610 000004
1565 006612 012704 021514
1566 006616 012705 006630
1567 006622 004737 021334
1568
1569 006626 000415
1570
1571 006630
1572 006630 177777 177777 177777
1573 006636 177777
1574 006640 000000 000000
1575 006644 000000 000000 177777
1576 006652 177777
1577 006654 047413 047604
1578 006660 000000
1579
1580
1581
1582
1583 006662 000004
1584 006664 012704 021514
1585 006670 012705 006702
1586 006674 004737 021334
1587
1588 006700 000415
1589
1590 006702
1591 006702 052525 052525 052525
1592 006710 052525
1593 006712 125252 125252
1594 006716 125252 125252 052525
1595 006724 052525
1596 006726 047547 047750
1597 006732 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 )

```

```

1598
1599
1600
1601
1602 006734 000004
1603 006736 012704 021514
1604 006742 012705 006754
1605 006746 004737 021334
1606
1607 006752 000415
1608
1609 006754
1610 006754 125252 125252 125252
1611 006762 125252
1612 006764 052525 052525
1613 006770 052525 052525 125252
1614 006776 125252
1615 007000 047417 047600
1616 007004 000000
1617
1618
1619
1620
1621 007006 000004
1622 007010 012704 021514
1623 007014 012705 007026
1624 007020 004737 021334
1625
1626 007024 000415
1627
1628 007026
1629 007026 177777 177777 177777
1630 007034 177777
1631 007036 100000 000000
1632 007042 100000 000000 177777
1633 007050 177777
1634 007052 043443 043654
1635 007056 000000
1636
1637
1638
1639
1640 007060 000004
1641 007062 012704 021514
1642 007066 012705 007100
1643 007072 004737 021334
1644
1645 007076 000415
1646
1647 007100
1648 007100 052525 000000 125252
1649 007106 177777
1650 007110 100000 000000
1651 007114 052525 000000 125252
1652 007122 177777
1653 007124 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

```



1654 007130 100014

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

1655

1656

1657

1658

1659

1660 007132 000004

1661 007134 012704 021536

1662 007140 012705 006556

1663 007144 004737 021334

1664

1665 007150 000400

1666

1667

1668

1669

1670

1671 007152 000004

1672 007154 012704 021536

1673 007160 012705 006630

1674 007164 004737 021334

1675

1676 007170 000400

1677

1678

1679

1680

1681

1682

1683 007172 000004

1684 007174 012704 021536

1685 007200 012705 006702

1686 007204 004737 021334

1687

1688 007210 000400

1689

1690

1691

1692

1693

1694 007212 000004

1695 007214 012704 021536

1696 007220 012705 006754

1697 007224 004737 021334

1698

1699 007230 000400

1700

1701

1702

1703

1704

1705

1706 007232 000004

1707 007234 012704 021536

\*\*\*\*\*  
; \*TEST 74 TEST FP AC1, F MODE, TEST PATTERN LDAF-1  
\*\*\*\*\*

TST74: SCOPE  
MOV #LDARF1,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 AC0 WITH TEST PATTERN LDAF-1, ABOVE )

\*\*\*\*\*  
; \*TEST 75 TEST FP AC1, F MODE, TEST PATTERN LDAF-2  
\*\*\*\*\*

TST75: SCOPE  
MOV #LDARF1,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 AC0 WITH TEST PATTERN LDAF-2, ABOVE )

\*\*\*\*\*  
; \*TEST 76 TEST FP AC1, F MODE, TEST PATTERN LDAF-3  
\*\*\*\*\*

TST76: SCOPE  
MOV #LDARF1,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 AC0 WITH TEST PATTERN LDAF-3, ABOVE )

\*\*\*\*\*  
; \*TEST 77 TEST FP AC1, F MODE, TEST PATTERN LDAF-4  
\*\*\*\*\*

TST77: SCOPE  
MOV #LDARF1,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 AC0 WITH TEST PATTERN LDAF-4, ABOVE )

\*\*\*\*\*  
; \*TEST 100 TEST FP AC1, F MODE, TEST PATTERN LDAF-5  
\*\*\*\*\*

TST100: SCOPE  
MOV #LDARF1,R4 ; PTR TO TESTING ROUTINE

F05

FPU BASIC INSTR TESTS MACY11 27(1006) 25-APR-77 09:12 PAGE 38  
 DQFPAB.P11 19-APR-77 13:36

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

SEQ 0042

1710 007240 012705 007026  
 1711 007244 004737 021334  
 1712  
 1713 007250 000400  
 1714  
 1715  
 1716  
 1717  
 1718  
 1719  
 1720 007252 000004  
 1721 007254 012704 021536  
 1722 007260 012705 007100  
 1723 007264 004737 021334  
 1724  
 1725 007270 000400  
 1726  
 1727  
 1728  
 1729  
 1730  
 1731  
 1732  
 1733 007272 000004  
 1734 007274 012704 021560  
 1735 007300 012705 006556  
 1736 007304 004737 021334  
 1737  
 1738 007310 000400  
 1739  
 1740  
 1741  
 1742  
 1743  
 1744  
 1745 007312 000004  
 1746 007314 012704 021560  
 1747 007320 012705 006630  
 1748 007324 004737 021334  
 1749  
 1750 007330 000400  
 1751  
 1752  
 1753  
 1754  
 1755  
 1756  
 1757 007332 000004  
 1758 007334 012704 021560  
 1759 007340 012705 006702  
 1760 007344 004737 021334  
 1761  
 1762 007350 000400  
 1763  
 1764  
 1765

```

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

BR TST101 ;;

; ( FOR DATA SEE TEST OF FP REG AC0 WITH TEST PATTERN LDAF-5, ABOVE )

;*****
;TEST 101 TEST FP AC1, F MODE, TEST PATTERN LDAF-6
;*****
TST101: SCOPE
MOV #LDAF1,R4 ; PTR TO TESTING ROUTINE
MOV #LDAF6,R5 ; PTR TO TEST DATA SET
JSR PC,@#LDAFT ; GO TEST

BR TST102 ;;

; ( FOR DATA SEE TEST OF FP REG AC0 WITH TEST PATTERN LDAF-6, ABOVE )

;*****
;TEST 102 TEST FP AC2, F MODE, TEST PATTERN LDAF-1
;*****
TST102: SCOPE
MOV #LDAF2,R4 ; PTR TO TESTING ROUTINE
MOV #LDAF1,R5 ; PTR TO TEST DATA SET
JSR PC,@#LDAFT ; GO TEST

BR TST103 ;;

; ( FOR DATA SEE TEST OF FP REG AC0 WITH TEST PATTERN LDAF-1, ABOVE )

;*****
;TEST 103 TEST FP AC2, F MODE, TEST PATTERN LDAF-2
;*****
TST103: SCOPE
MOV #LDAF2,R4 ; PTR TO TESTING ROUTINE
MOV #LDAF2,R5 ; PTR TO TEST DATA SET
JSR PC,@#LDAFT ; GO TEST

BR TST104 ;;

; ( FOR DATA SEE TEST OF FP REG AC0 WITH TEST PATTERN LDAF-2, ABOVE )

;*****
;TEST 104 TEST FP AC2, F MODE, TEST PATTERN LDAF-3
;*****
TST104: SCOPE
MOV #LDAF2,R4 ; PTR TO TESTING ROUTINE
MOV #LDAF3,R5 ; PTR TO TEST DATA SET
JSR PC,@#LDAFT ; GO TEST

BR TST105 ;;

; ( FOR DATA SEE TEST OF FP REG AC0 WITH TEST PATTERN LDAF-3, ABOVE )

```

```

1766
1767
1768
1769 007352 000004
1770 007354 012704 021560
1771 007360 012705 006754
1772 007364 004737 021334
1773
1774 007370 000400
1775
1776
1777
1778
1779
1780
1781 007372 000004
1782 007374 012704 021560
1783 007400 012705 007026
1784 007404 004737 021334
1785
1786 007410 000400
1787
1788
1789
1790
1791
1792
1793 007412 000004
1794 007414 012704 021560
1795 007420 012705 007100
1796 007424 004737 021334
1797
1798 007430 000400
1799
1800
1801
1802
1803
1804
1805
1806 007432 000004
1807 007434 012704 021602
1808 007440 012705 006556
1809 007444 004737 021334
1810
1811 007450 000400
1812
1813
1814
1815
1816
1817
1818 007452 000004
1819 007454 012704 021602
1820 007460 012705 006630
1821 007464 004737 021334

```

```

*****
*TEST 105 TEST FP AC2, F MODE, TEST PATTERN LDAF-4
*****
TST105: SCOPE
MOV #LDARF2,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 ACO WITH TEST PATTERN LDAF-4, ABOVE )
*****
*TEST 106 TEST FP AC2, F MODE, TEST PATTERN LDAF-5
*****
TST106: SCOPE
MOV #LDARF2,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 ACO WITH TEST PATTERN LDAF-5, ABOVE )
*****
*TEST 107 TEST FP AC2, F MODE, TEST PATTERN LDAF-6
*****
TST107: SCOPE
MOV #LDARF2,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 ACO WITH TEST PATTERN LDAF-6, ABOVE )
*****
*TEST 110 TEST FP AC3, F MODE, TEST PATTERN LDAF-1
*****
TST110: SCOPE
MOV #LDARF3,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 ACO WITH TEST PATTERN LDAF-1, ABOVE )
*****
*TEST 111 TEST FP AC3, F MODE, TEST PATTERN LDAF-2
*****
TST111: SCOPE
MOV #LDARF3,R4 ; PTR TO TESTING ROUTINE
MOV #LDAF2,R5 ; PTR TO TEST DATA SET
JSR PC,@#LDAFT ; GO TEST

```

```

1822
1823 007470 000400
1824
1825
1826
1827
1828
1829
1830 007472 000004
1831 007474 012704 021602
1832 007500 012705 006702
1833 007504 004737 021334
1834
1835 007510 000400
1836
1837
1838
1839
1840
1841
1842 007512 000004
1843 007514 012704 021602
1844 007520 012705 006754
1845 007524 004737 021334
1846
1847 007530 000400
1848
1849
1850
1851
1852
1853
1854 007532 000004
1855 007534 012704 021602
1856 007540 012705 007026
1857 007544 004737 021334
1858
1859 007550 000400
1860
1861
1862
1863
1864
1865
1866 007552 000004
1867 007554 012704 021602
1868 007560 012705 007100
1869 007564 004737 021334
1870
1871 007570 000400
1872
1873
1874
1875
1876
1877

```

```

BR TST112 ;;
; ( FOR DATA SEE TEST OF FP REG AC0 WITH TEST PATTERN LDAF-2, ABOVE )
;*****
;TEST 112 TEST FP AC3, F MODE, TEST PATTERN LDAF-3
;*****
†TST112: SCOPE
MOV #LDAF3,R4 ; PTR TO TESTING ROUTINE
MOV #LDAF3,R5 ; PTR TO TEST DATA SET
JSR PC,@#LDAFT ; GO TEST
BR TST113 ;;
; ( FOR DATA SEE TEST OF FP REG AC0 WITH TEST PATTERN LDAF-3, ABOVE )
;*****
;TEST 113 TEST FP AC3, F MODE, TEST PATTERN LDAF-4
;*****
†TST113: SCOPE
MOV #LDAF3,R4 ; PTR TO TESTING ROUTINE
MOV #LDAF4,R5 ; PTR TO TEST DATA SET
JSR PC,@#LDAFT ; GO TEST
BR TST114 ;;
; ( FOR DATA SEE TEST OF FP REG AC0 WITH TEST PATTERN LDAF-4, ABOVE )
;*****
;TEST 114 TEST FP AC3, F MODE, TEST PATTERN LDAF-5
;*****
†TST114: SCOPE
MOV #LDAF3,R4 ; PTR TO TESTING ROUTINE
MOV #LDAF5,R5 ; PTR TO TEST DATA SET
JSR PC,@#LDAFT ; GO TEST
BR TST115 ;;
; ( FOR DATA SEE TEST OF FP REG AC0 WITH TEST PATTERN LDAF-5, ABOVE )
;*****
;TEST 115 TEST FP AC3, F MODE, TEST PATTERN LDAF-6
;*****
†TST115: SCOPE
MOV #LDAF3,R4 ; PTR TO TESTING ROUTINE
MOV #LDAF6,R5 ; PTR TO TEST DATA SET
JSR PC,@#LDAFT ; GO TEST
BR TST116 ;;
; ( FOR DATA SEE TEST OF FP REG AC0 WITH TEST PATTERN LDAF-6, ABOVE )
;*****
;TEST 116 TEST FP AC4, F MODE, TEST PATTERN LDAF-1

```

```

1878
1879 007572 000004
1880 007574 012704 021624
1881 007600 012705 006556
1882 007604 004737 021334
1883
1884 007610 000400
1885
1886
1887
1888
1889
1890
1891 007612 000004
1892 007614 012704 021624
1893 007620 012705 006630
1894 007624 004737 021334
1895
1896 007630 000400
1897
1898
1899
1900
1901
1902
1903 007632 000004
1904 007634 012704 021624
1905 007640 012705 006702
1906 007644 004737 021334
1907
1908 007650 000400
1909
1910
1911
1912
1913
1914
1915 007652 000004
1916 007654 012704 021624
1917 007660 012705 006754
1918 007664 004737 021334
1919
1920 007670 000400
1921
1922
1923
1924
1925
1926
1927 007672 000004
1928 007674 012704 021624
1929 007700 012705 007026
1930 007704 004737 021334
1931
1932 007710 000400
1933

```

```

*****
↑ST116: SCOPE
MOV #LDAF4,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 AC0 WITH TEST PATTERN LDAF-1, ABOVE )

*****
; *TEST 117 TEST FP AC4, F MODE, TEST PATTERN LDAF-2
*****
↑ST117: SCOPE
MOV #LDAF4,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 AC0 WITH TEST PATTERN LDAF-2, ABOVE )

*****
; *TEST 120 TEST FP AC4, F MODE, TEST PATTERN LDAF-3
*****
↑ST120: SCOPE
MOV #LDAF4,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 AC0 WITH TEST PATTERN LDAF-3, ABOVE )

*****
; *TEST 121 TEST FP AC4, F MODE, TEST PATTERN LDAF-4
*****
↑ST121: SCOPE
MOV #LDAF4,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 AC0 WITH TEST PATTERN LDAF-4, ABOVE )

*****
; *TEST 122 TEST FP AC4, F MODE, TEST PATTERN LDAF-5
*****
↑ST122: SCOPE
MOV #LDAF4,R4 ; PTR TO TESTING ROUTINE
MOV #LDAF5,R5 ; PTR TO TEST DATA SET
JSR PC,@#LDAFT ; GO TEST

BR TST123 ;;

```

```

1934
1935
1936
1937
1938
1939
1940 007712 000004
1941 007714 012704 021654
1942 007720 012705 006556
1943 007724 004737 021334
1944
1945 007730 000400
1946
1947
1948
1949
1950
1951
1952 007732 000004
1953 007734 012704 021654
1954 007740 012705 006630
1955 007744 004737 021334
1956
1957 007750 000400
1958
1959
1960
1961
1962
1963
1964 007752 000004
1965 007754 012704 021654
1966 007760 012705 006702
1967 007764 004737 021334
1968
1969 007770 000400
1970
1971
1972
1973
1974
1975
1976 007772 000004
1977 007774 012704 021654
1978 010000 012705 006754
1979 010004 004737 021334
1980
1981 010010 000400
1982
1983
1984
1985
1986
1987
1988 010012 000004
1989 010014 012704 021654

```

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

```

*****
;*TEST 123 TEST FP AC5, 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 )

```

*****
;*TEST 124 TEST FP AC5, 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 )

```

*****
;*TEST 125 TEST FP AC5, 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 )

```

*****
;*TEST 126 TEST FP AC5, 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 )

```

*****
;*TEST 127 TEST FP AC5, F MODE, TEST PATTERN LDAF-5
*****

```

```

†TST127: SCOPE
          MOV     #LDARF5,R4      ; PTR TO TESTING ROUTINE

```

K05

FPU BASIC INSTR TESTS MACY11 27(1006) 25-APR-77 09:12 PAGE 43  
DQFPAB.P11 19-APR-77 13:36 T127 TEST FP ACS, F MODE, TEST PATTERN LDAF-5

SEQ 0047

1990 010020 012705 007026  
1991 010024 004737 021334  
1992  
1993 010030 000400  
1994  
1995  
1996  
1997  
1998

MOV #LDAF5, R5 ; PTR TO TEST DATA SET  
JSR PC, @LDAF5 ; GO TEST  
BR TST130 ;

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

```

1999
2000
2001
2002
2003
2004
2005
2006
2007 010032 000004
2008 010034 170127 047600
2009 010040 016767 000102 171122
2010 010046 016767 000076 171116
2011 010054 016767 000072 171112
2012 010062 016767 000066 171106
2013
2014 010070 012705 001170
2015 010074 172415
2016
2017 010076 174067 171076
2018
2019 010102 026767 171062 171070
2020 010110 001014
2021 010112 026767 171054 171062
2022 010120 001010
2023 010122 026767 171046 171054
2024 010130 001004
2025 010132 026767 171040 171046
2026 010140 001401
2027 010142 104031
2028
2029 010144
2030 010144 000404
2031
2032 010146 052525 052525 052525
2033 010154 052525
2034
2035
2036
2037
2038
2039 010156 000004
2040 010160 170127 047600
2041 010164 016767 000112 170776
2042 010172 016767 000106 170772
2043 010200 016767 000102 170766
2044 010206 016767 000076 170762
2045
2046 010214 012704 001200
2047 010220 172544
2048
2049 010222 174137 001200
2050
2051 010226 020427 001170
2052 010232 001401
2053 010234 104032
2054 010236

```

```

;*****
;.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 #$REG0,R5 ; ADDR(DATA)
LDD (R5),AC0 ; M1-R5
STD AC0,$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 ;;
ADRD1: .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 #$REG0+10,R4 ; ADDR(DATA+10)
LDD -(R4),AC1 ; M4-R4
STD AC1,2#$REG4 ; M3-R7
CMP R4,$REG0 ; WAS FSRC ADDR REG DECRE RIGHT AMOUNT?
BEQ 66$
66$: ERROR 32 ; NOT EQUAL, SIGNAL ERROR
66$:

```



```

2055
2056 010236 026767 170726 170734      CMP      $REG0,$REG4      ; 1ST WORD PATTERN CHECK?
2057 010244 001014                      BNE      64$              ;
2058 010246 026767 170720 170726      CMP      $REG1,$REG5      ; 2ND WORD PATTERN CHECK?
2059 010254 001010                      BNE      64$              ;
2060 010256 026767 170712 170720      CMP      $REG2,$REG6      ; 3RD WORD PATTERN CHECK?
2061 010264 001004                      BNE      64$              ;
2062 010266 026767 170704 170712      CMP      $REG3,$REG7      ; 4TH WORD PATTERN CHECK?
2063 010274 001401                      BEQ      65$              ;
2064 010276 104033                      64$:   ERROR      33      ; PATTERN DOESNT MATCH
2065
2066 010300                      65$:
2067 010300 000404                      BR       TST132          ;;
2068
2069 010302 170360 170360 170360  ADDR2: .WORD  ALT4N,ALT4N,ALT4N,ALT4N ; TEST PATTERN
2070 010310 170360
2071
2072
2073
2074
2075
2076 010312 000004
2077 010314 170127 047600
2078 010320 016767 000132 170642      MOV      ADDR3+0,$REG0    ; D MODE FPS
2079 010326 016767 000126 170636      MOV      ADDR3+2,$REG1    ; MOVE PATTERN
2080 010334 016767 000122 170632      MOV      ADDR3+4,$REG2    ;
2081 010342 016767 000116 170626      MOV      ADDR3+6,$REG3    ;
2082
2083 010350 012767 001170 170632      MOV      #$REG0,$REG10   ; ADDR(DATA)
2084 010356 012705 001214                      MOV      #$REG10+4,R5    ; ADDR(ADDR(DATA))+4
2085 010362 172675 177774                      LDD      2-4(R5),AC2     ; M7-R5
2086
2087 010366 012767 001200 170616      MOV      #$REG4,$REG11   ; ADDR(DEST)
2088 010374 012704 001212                      MOV      #$REG11,R4      ; ADDR(ADDR(DEST))
2089 010400 174234                      STD      AC2,2(R4)+      ; M3-R4
2090
2091 010402 020427 001214                      CMP      R4,$REG11+2     ; WAS FDST ADDR REG INCRE RIGHT AMOUNT?
2092 010406 001401                      BEQ      66$              ;
2093 010410 104034                      66$:   ERROR      34      ; NOT EQUAL, SIGNAL ERROR
2094 010412
2095
2096 010412 026767 170552 170560      CMP      $REG0,$REG4      ; 1ST WORD PATTERN CHECK?
2097 010420 001014                      BNE      64$              ;
2098 010422 026767 170544 170552      CMP      $REG1,$REG5      ; 2ND WORD PATTERN CHECK?
2099 010430 001010                      BNE      64$              ;
2100 010432 026767 170536 170544      CMP      $REG2,$REG6      ; 3RD WORD PATTERN CHECK?
2101 010440 001004                      BNE      64$              ;
2102 010442 026767 170530 170536      CMP      $REG3,$REG7      ; 4TH WORD PATTERN CHECK?
2103 010450 001401                      BEQ      65$              ;
2104 010452 104035                      64$:   ERROR      35      ; PATTERN DOESNT MATCH
2105
2106 010454                      65$:
2107 010454 000404                      BR       TST133          ;;
2108
2109 010456 125252 125252 125252  ADDR3: .WORD  ALTN,ALTN,ALTN,ALTN ; TEST PATTERN
2110 010464 125252

```

```

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

```

```

TST132: SCOPE
LDFPS #047600 ; D MODE FPS
MOV ADDR3+0,$REG0 ; MOVE PATTERN
MOV ADDR3+2,$REG1 ;
MOV ADDR3+4,$REG2 ;
MOV ADDR3+6,$REG3 ;
MOV #$REG0,$REG10 ; ADDR(DATA)
MOV #$REG10+4,R5 ; ADDR(ADDR(DATA))+4
LDD 2-4(R5),AC2 ; M7-R5
MOV #$REG4,$REG11 ; ADDR(DEST)
MOV #$REG11,R4 ; ADDR(ADDR(DEST))
STD AC2,2(R4)+ ; M3-R4
CMP R4,$REG11+2 ; WAS FDST ADDR REG INCRE RIGHT AMOUNT?
BEQ 66$ ;
ERROR 34 ; NOT EQUAL, SIGNAL ERROR
66$:

```

N05

FPU BASIC INSTR TESTS MACY11 27(1006) 25-APR-77 09:12 PAGE 46  
 DQFPAB.P11 19-APR-77 13:36

T132 TEST OF LOAD-FSRC MODE-7, STORE-FDST MODE-3, D MODE

SEQ 0050

```

2111
2112
2113
2114
2115
2116 010466 000004
2117 010470 170127 047600
2118 010474 016767 000112 170466
2119 010502 016767 000106 170462
2120 010510 016767 000102 170456
2121 010516 016767 000076 170452
2122
2123 010524 172767 170440
2124
2125 010530 012705 001200
2126 010534 174325
2127
2128 010536 020527 001210
2129 010542 001401
2130 010544 104036
2131 010546
2132
2133 010546 026767 170416 170424
2134 010554 001014
2135 010556 026767 170410 170416
2136 010564 001010
2137 010566 026767 170402 170410
2138 010574 001004
2139 010576 026767 170374 170402
2140 010604 001401
2141 010606 104031
2142
2143 010610
2144 010610 000404
2145
2146 010612 007417 007417 007417
2147 010620 007417
2148
2149
2150
2151
2152
2153 010622 000004
2154 010624 170127 047600
2155 010630 016767 000114 170332
2156 010636 016767 000110 170326
2157 010644 016767 000104 170322
2158 010652 016767 000100 170316
2159
2160 010660 012700 001170
2161 010664 172420
2162
2163 010666 012703 001200
2164 010672 174013
2165
2166 010674 020027 001200
    
```

```

*****
; *TEST 133 TEST OF LOAD-FSRC MODE-6(PC), STORE-FDST MODE-2, D MODE
*****
TST133: SCOPE
LDFPS #047600 ; D MODE FPS
MOV ADRD4+0,$REG0 ; MOVE PATTERN
MOV ADRD4+2,$REG1
MOV ADRD4+4,$REG2
MOV ADRD4+6,$REG3
LDD $REG0,AC3 ; M6-R7
MOV #SREG4,R5 ; ADDR(DEST)
STD AC3,(R5)+ ; M2-R5
CMP R5,#SREG4+10 ; WAS FDST ADDR REG INCRE RIGHT AMOUNT?
BEQ 66$
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 ;;
ADR4: .WORD ALT4P,ALT4P,ALT4P,ALT4P ; TEST PATTERN
    
```

```

*****
; *TEST 134 TEST OF LOAD-FSRC MODE-2, STORE-FDST MODE-1, D MODE
*****
TST134: SCOPE
LDFPS #047600 ; D MODE FPS
MOV ADRD5+0,$REG0 ; MOVE PATTERN
MOV ADRD5+2,$REG1
MOV ADRD5+4,$REG2
MOV ADRD5+6,$REG3
MOV #SREG0,R0 ; ADDR(DATA)
LDD (R0)+,AC0 ; M6-R0
MOV #SREG4,R3 ; ADDR(DEST)
STD AC0,(R3) ; M1-R3
CMP R0,#SREG0+10 ; WAS FSRC ADDR REG INCRE RIGHT AMOUNT?
    
```

```

2167 010700 001401      BEQ      66$      ;
2168 010702 104037      ERROR    37      ; NOT EQUAL, SIGNAL ERROR
2169 010704      66$:
2170
2171 010704 026767 170260 170266      CMP      $REG0,$REG4 ; 1ST WORD PATTERN CHECK?
2172 010712 001014      BNE      64$      ;
2173 010714 026767 170252 170260      CMP      $REG1,$REG5 ; 2ND WORD PATTERN CHECK?
2174 010722 001010      BNE      64$      ;
2175 010724 026767 170244 170252      CMP      $REG2,$REG6 ; 3RD WORD PATTERN CHECK?
2176 010732 001004      BNE      64$      ;
2177 010734 026767 170236 170244      CMP      $REG3,$REG7 ; 4TH WORD PATTERN CHECK?
2178 010742 001401      BEQ      65$      ;
2179 010744 104040      ERROR    40      ; PATTERN DOESNT MATCH
2180
2181 010746      65$:
2182 010746 000404      BR       TST135   ;;
2183
2184 010750 125252 125252 125252  ADDR5: .WORD  ALTN,ALTN,ALTN,ALTN ; TEST PATTERN
2185 010756 125252
2186
2187
2188
2189
2190
2191 010760 000004      *TEST 135  TEST OF LOAD-FSRC MODE-5, STORE-FDST MODE-7(PC), D MODE
2192 010762 170127 047600      *TEST135: SCOPE
2193 010766 016767 000126 170174      LDFPS   #047600 ; D MODE FPS
2194 010774 016767 000122 170170      MOV     ADDR6+0,$REG0 ; MOVE PATTERN
2195 011002 016767 000116 170164      MOV     ADDR6+2,$REG1 ;
2196 011010 016767 000112 170160      MOV     ADDR6+4,$REG2 ;
2197
2198 011016 012767 001170 170164      MOV     ADDR6+6,$REG3 ;
2199 011024 012701 001212      MOV     #SREG0,$REG10 ; ADDR(DATA)
2200 011030 172551      MOV     #SREG10+2,R1 ; ADDR(ADDR(DATA)+2)
2201      LDD    2-(R1),AC1 ; M5-R1
2202 011032 012767 001200 170152      MOV     #SREG4,$REG11 ; ADDR(DEST)
2203 011040 174177 170146      STD     AC1,2SREG11 ; M7-R7
2204
2205 011044 020127 001210      CMP     R1,#SREG10 ; WAS FSRC ADDR REG DECRE RIGHT AMOUNT?
2206 011050 001401      BEQ     66$      ;
2207 011052 104041      ERROR   41      ; NOT EQUAL, SIGNAL ERROR
2208 011054      66$:
2209
2210 011054 026767 170110 170116      CMP     $REG0,$REG4 ; 1ST WORD PATTERN CHECK?
2211 011062 001014      BNE     64$      ;
2212 011064 026767 170102 170110      CMP     $REG1,$REG5 ; 2ND WORD PATTERN CHECK?
2213 011072 001010      BNE     64$      ;
2214 011074 026767 170074 170102      CMP     $REG2,$REG6 ; 3RD WORD PATTERN CHECK?
2215 011102 001004      BNE     64$      ;
2216 011104 026767 170066 170074      CMP     $REG3,$REG7 ; 4TH WORD PATTERN CHECK?
2217 011112 001401      BEQ     65$      ;
2218 011114 104042      ERROR   42      ; PATTERN DOESNT MATCH
2219
2220 011116      65$:
2221 011116 000404      BR       TST136   ;;
2222

```

```

2223 011120 007417 007417 007417 ADRD6: .WORD ALT4P,ALT4P,ALT4P,ALT4P ; TEST PATTERN
2224 011126 007417
2225
2226
2227
2228 ;*****
;#TEST 136 TEST OF LOAD-FSRC MODE-2(PC), STORE-FDST MODE-7, D MODE
;*****
2229 †TST136: SCOPE
2230 011130 000004 LDFPS #047600 ; D MODE FPS
2231 011132 170127 047600 MOV ADRD7+0,$REG0 ; MOVE PATTERN
2232 011136 016767 000130 170024 MOV ADRD7+2,$REG1 ;
2233 011144 016767 000124 170020 MOV ADRD7+4,$REG2 ;
2234 011152 016767 000120 170014 MOV ADRD7+6,$REG3 ;
2235 011160 016767 000114 170010
2236
2237 011166 016767 167776 000002 MOV $REG0,1$ ; PUT DATA
2238 011174 172627 LDD (PC)+,AC2 ; M2-R7
2239 011176 000000 1$: .WORD 0 ; DATA
2240 011200 000403 BR 2$ ; THIS SHOULD BE NEXT INSTR AFTER FPP PC MODE 2
2241 011202 104120 ERROR 120 ; NOT HERE (+4 INCRE)
2242 011204 104121 ERROR 121 ; OR HERE (+6 INCRE)
2243 011206 104122 ERROR 122 ; OR HERE (+10 INCRE)
2244
2245 011210 012767 001200 167772 2$: MOV #$REG4,$REG10 ; ADDR(DEST)
2246 011216 012700 001152 MOV #$REG10-36,R0 ; ADDR(ADDR(DEST))-36
2247 011222 174270 000036 STD AC2,36(R0) ; M7-R0
2248
2249 011226 026767 167736 167744 CMP $REG0,$REG4 ; 1ST WORD PATTERN CHECK?
2250 011234 001014 BNE 64$ ;
2251 011236 026767 167730 167736 CMP $REG1,$REG5 ; 2ND WORD PATTERN CHECK?
2252 011244 001010 BNE 64$ ;
2253 011246 026767 167722 167730 CMP $REG2,$REG6 ; 3RD WORD PATTERN CHECK?
2254 011254 001004 BNE 64$ ;
2255 011256 026767 167714 167722 CMP $REG3,$REG7 ; 4TH WORD PATTERN CHECK?
2256 011264 001401 BEQ 65$ ;
2257 011266 104043 64$: ERROR 43 ; PATTERN DOESNT MATCH
2258
2259 011270 65$: BR TST137 ;
2260 011270 000404
2261
2262 011272 104117 000000 000000 ADRD7: .WORD ERROR117,0,0,0 ; TEST PATTERN
2263 011300 000000
2264
2265
2266 ;*****
;#TEST 137 TEST OF LOAD-FSRC MODE-7(PC), STORE-FDST MODE-6, D MODE
;*****
2267 †TST137: SCOPE
2268 011302 000004 LDFPS #047600 ; D MODE FPS
2269 011304 170127 047600 MOV ADRD10+0,$REG0 ; MOVE PATTERN
2270 011310 016767 000112 167652 MOV ADRD10+2,$REG1 ;
2271 011316 016767 000106 167646 MOV ADRD10+4,$REG2 ;
2272 011324 016767 000102 167642 MOV ADRD10+6,$REG3 ;
2273 011332 016767 000076 167636
2274
2275 011340 012767 001170 167642 MOV #$REG0,$REG10 ; ADDR(DATA)
2276 011346 172777 167636 LDD 3$REG10,AC3 ; M7-R7
2277
2278

```

```

2279 011352 012704 001212      MOV      #SREG4+12,R4      ; ADDR(DEST)+12
2280 011356 174364 177766      STD      AC3,-12(R4)      ; M6-R4
2281
2282 011362 026767 167602 167610      CMP      $REG0,$REG4      ; 1ST WORD PATTERN CHECK?
2283 011370 001014 64$
2284 011372 026767 167574 167602      CMP      $REG1,$REG5      ; 2ND WORD PATTERN CHECK?
2285 011400 001010 64$
2286 011402 026767 167566 167574      CMP      $REG2,$REG6      ; 3RD WORD PATTERN CHECK?
2287 011410 001004 64$
2288 011412 026767 167560 167566      CMP      $REG3,$REG7      ; 4TH WORD PATTERN CHECK?
2289 011420 001401 65$
2290 011422 104044 64$:      BEQ      ERROR 44      ; PATTERN DOESNT MATCH
2291
2292 011424 65$:
2293 011424 000404      BR      TST140      ;;
2294
2295 011426 170360 170360 170360  ADDR10: .WORD  ALT4N,ALT4N,ALT4N,ALT4N ; TEST PATTERN
2296 011434 170360
2297
2298
2299
2300
2301
2302 011436 000004
2303 011440 170127 047600
2304 011444 016767 000114 167516      MOV      ADDR11+0,$REG0      ; D MODE FPS
2305 011452 016767 000110 167512      MOV      ADDR11+2,$REG1      ; MOVE PATTERN
2306 011460 016767 000104 167506      MOV      ADDR11+4,$REG2
2307 011466 016767 000100 167502      MOV      ADDR11+6,$REG3
2308
2309 011474 012767 001170 167506      MOV      #SREG0,$REG10      ; ADDR(DATA)
2310 011502 012702 001210      MOV      #SREG10,R2      ; ADDR(ADDR(DATA))
2311 011506 172432      LDD     3(R2)+,AC0      ; M3-R2
2312
2313 011510 012767 104117 000002      MOV      #ERROR117,1$      ; SETUP ERROR CALL FOR WRONG INCREMENT
2314 011516 174027      STD     AC0,(PC)+      ; M2-R7
2315 011520 000000 1$:      .WORD  0      ; DEST
2316 011522 000403      BR      2$      ; THIS SHOULD BE NEXT INSTR AFTER FPP PC MODE 2
2317 011524 104120      ERROR  120      ; NOT HERE (+4 INCRE)
2318 011526 104121      ERROR  121      ; OR HERE (+6 INCRE)
2319 011530 104122      ERROR  122      ; OR HERE (+10 INCRE)
2320 011532 016767 177762 167440 2$:      MOV      1$,SREG4      ; STORE DEST
2321
2322 011540 020227 001212      CMP      R2,#SREG10+2      ; WAS FSRC ADDR REG INCRE BY RIGHT AMOUNT?
2323 011544 001401 64$
2324 011546 104045      BEQ     ERROR 45      ; NOT EQUAL, SIGNAL ERROR
2325 011550 64$:
2326
2327 011550 026767 167414 167422      CMP      $REG0,$REG4      ; WAS 1 WORD STORED OK?
2328 011556 001401 65$
2329 011560 104046      BEQ     ERROR 46      ; NOT EQUAL, SIGNAL ERROR
2330 011562 65$:
2331
2332 011562 000404      BR      TST141      ;;
2333
2334 011564 052525 000000 000000  ADDR11: .WORD  ALTP,0,0,0      ; TEST PATTERN
    
```

```

*****
; *TEST 140 TEST OF LOAD-FSRC MODE-3, STORE-FDST MODE-2(PC), D MODE
*****
TST140: SCOPE
    
```

2335 011572 000000

2336  
2337

2338

2339

2340

2341 011574 000004

2342 011576 170127 047600

2343 011602 016767 000116 167360

2344 011610 016767 000112 167354

2345 011616 016767 000106 167350

2346 011624 016767 000102 167344

2347

2348 011632 012703 001262

2349 011636 172563 177706

2350

2351 011642 012701 001210

2352 011646 174141

2353

2354 011650 020127 001200

2355 011654 001401

2356 011656 104047

2357 011660

2358

2359 011660 026767 167304 167312

2360 011666 001014

2361 011670 026767 167276 167304

2362 011676 001010

2363 011700 026767 167270 167276

2364 011706 001004

2365 011710 026767 167262 167270

2366 011716 001401

2367 011720 104050

2368

2369 011722

2370 011722 000404

2371

2372 011724 170360 170360 170360

2373 011732 170360

2374

2375

2376

2377

2378

2379 011734 000004

2380 011736 170127 047600

2381 011742 016767 000102 167220

2382 011750 016767 000076 167214

2383 011756 016767 000072 167210

2384 011764 016767 000066 167204

2385

2386 011772 172637 001170

2387

2388 011776 174203

2389 012000 174367 167174

2390

```

*****
; *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 #SREG0+72,R3 ; ADDR(DATA)+72
LDD -72(R3),AC1 ; M6-R3
MOV #SREG4+10,R1 ; ADDR(DEST+10)
STD AC1,-(R1) ; M4-R1
CMP R1,#SREG4 ; WAS FDST ADDR REG DECRE RIGHT AMOUNT?
BEQ 66$ ;
ERROR 47 ; NOT EQUAL, SIGNAL ERROR

```

66\$:

```

CMP SREG0,$REG4 ; 1ST WORD PATTERN CHECK?
BNE 64$ ;
CMP SREG1,$REG5 ; 2ND WORD PATTERN CHECK?
BNE 64$ ;
CMP SREG2,$REG6 ; 3RD WORD PATTERN CHECK?
BNE 64$ ;
CMP SREG3,$REG7 ; 4TH WORD PATTERN CHECK?
BEQ 65$ ;
ERROR 50 ; PATTERN DOESNT MATCH

```

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 @SREG0,AC2 ; M3-R7
STD AC2,AC3 ; M0-R3
STD AC3,$REG4 ; M6-R7 WORKS

```

```

2391 012004 026767 167160 167166      CMP      $REG0,$REG4      ; 1ST WORD PATTERN CHECK?
2392 012012 001014                BNE      64$              ;
2393 012014 026767 167152 167160      CMP      $REG1,$REG5      ; 2ND WORD PATTERN CHECK?
2394 012022 001010                BNE      64$              ;
2395 012024 026767 167144 167152      CMP      $REG2,$REG6      ; 3RD WORD PATTERN CHECK?
2396 012032 001004                BNE      64$              ;
2397 012034 026767 167136 167144      CMP      $REG3,$REG7      ; 4TH WORD PATTERN CHECK?
2398 012042 001401                BEQ      65$              ;
2399 012044 104051                ERROR    51                ; PATTERN DOESNT MATCH

```

```

2400
2401 012046                65$:
2402 012046 000404                BR      TST143            ;;
2403
2404 012050 125252 125252 125252  ADDR13: .WORD  ALTN,ALTN,ALTN,ALTN      ; TEST PATTERN
2405 012056 125252
2406
2407
2408

```

```

;*****
; *TEST 143      TEST OF LOAD-FSRC MODE-0, STORE-FDST MODE-5, D MODE
;*****

```

```

TST143: SCOPE
2409
2410
2411 012060 000004                LDFPS   #047600           ; D MODE FPS
2412 012062 170127 047600      MOV     ADDR14+0,$REG0    ; MOVE PATTERN
2413 012066 016767 000122 167074  MOV     ADDR14+2,$REG1
2414 012074 016767 000116 167070  MOV     ADDR14+4,$REG2
2415 012102 016767 000112 167064  MOV     ADDR14+6,$REG3
2416 012110 016767 000106 167060
2417
2418 012116 172467 167046      LDD     $REG0,AC0         ; M6-R7 WORKS
2419 012122 172500                LDD     AC0,AC1          ; M0-R0
2420
2421 012124 012767 001200 167056  MOV     #$REG4,$REG10    ; ADDR(DEST)
2422 012132 012702 001212      MOV     #$REG10+2,R2     ; ADDR(ADDR(DEST)+2)
2423 012136 174152                STD     AC1,2-(R2)       ; M5-R2
2424
2425 012140 020227 001210      CMP     R2,$REG10        ; WAS FDST ADDR REG DECRE RIGHT AMOUNT?
2426 012144 001401                BEQ     66$              ;
2427 012146 104052                ERROR    52                ; NOT EQUAL, SIGNAL ERROR
2428 012150                66$:

```

```

2429
2430 012150 026767 167014 167022      CMP     $REG0,$REG4      ; 1ST WORD PATTERN CHECK?
2431 012156 001014                BNE     64$              ;
2432 012160 026767 167006 167014      CMP     $REG1,$REG5      ; 2ND WORD PATTERN CHECK?
2433 012166 001010                BNE     64$              ;
2434 012170 026767 167000 167006      CMP     $REG2,$REG6      ; 3RD WORD PATTERN CHECK?
2435 012176 001004                BNE     64$              ;
2436 012200 026767 166772 167000      CMP     $REG3,$REG7      ; 4TH WORD PATTERN CHECK?
2437 012206 001401                BEQ     65$              ;
2438 012210 104046                ERROR    46                ; PATTERN DOESNT MATCH

```

```

2439
2440 012212                65$:
2441 012212 000404                BR      TST144            ;;
2442
2443 012214 125252 125252 125252  ADDR14: .WORD  ALTN,ALTN,ALTN,ALTN      ; TEST PATTERN
2444 012222 125252
2445
2446

```

```

2447
2448
2449
2450 012224 000004
2451 012226 170127 047400
2452 012232 016767 000060 166730
2453 012240 016767 000054 166724
2454
2455 012246 012702 001170
2456 012252 172412
2457
2458 012254 012705 001200
2459 012260 174045
2460
2461 012262 020527 001174
2462 012266 001401
2463 012270 104053
2464 012272
2465
2466 012272 026767 166672 166674
2467 012300 001004
2468 012302 026767 166664 166666
2469 012310 001401
2470 012312 104054
2471
2472 012314
2473 012314 000402
2474
2475 012316 170360 170360
2476
2477
2478
2479
2480
2481 012322 000004
2482 012324 170127 047400
2483 012330 016767 000070 166632
2484 012336 016767 000064 166626
2485
2486 012344 012767 001170 166626
2487 012352 012704 001200
2488 012356 172534
2489
2490 012360 012700 001166
2491 012364 174160 000006
2492
2493 012370 020427 001202
2494 012374 001401
2495 012376 104055
2496 012400
2497
2498 012400 026767 166564 166566
2499 012406 001004
2500 012410 026767 166556 166560
2501 012416 001401
2502 012420 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 DECREASE 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 2(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 INCREASE 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

```



```

2503
2504 012422
2505 012422 000402
2506
2507 012424 052525 052525
2508
2509
2510
2511
2512
2513 012430 000004
2514 012432 170127 047400
2515 012436 016767 000104 166524
2516 012444 016767 000100 166520
2517
2518 012452 012767 001170 166520
2519 012460 012700 001202
2520 012464 172650
2521
2522 012466 012767 001174 166506
2523 012474 012702 001202
2524 012500 174232
2525
2526 012502 020027 001200
2527 012506 001401
2528 012510 104057
2529 012512
2530
2531 012512 020227 001204
2532 012516 001401
2533 012520 104060
2534 012522
2535
2536 012522 026767 166442 166444
2537 012530 001004
2538 012532 026767 166434 166436
2539 012540 001401
2540 012542 104061
2541
2542 012544
2543 012544 000402
2544
2545 012546 007417 007417
2546
2547
2548
2549
2550
2551 012552 000004
2552 012554 170127 047400
2553 012560 016767 000060 166402
2554 012566 016767 000054 166376
2555
2556 012574 012767 001170 166376
2557 012602 012702 001162
2558 012606 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,$REG4 ; ADDR(DATA)
MOV #SREG4+2,R0 ; ADDR(ADDR(DATA)+2)
LDF 2-(R0),AC2 ; M5-R0
MOV #SREG2,$REG5 ; ADDR(DEST)
MOV #SREG5,R2 ; ADDR(ADDR(DEST))
STF AC2,2(R2)+ ; M3-R2
CMP R0,#SREG4 ; WAS FSRC ADDR REG DECRE RIGHT AMOUNT?
BEQ 66$ ;
ERROR 57 ; NOT EQUAL, SIGNAL ERROR
66$:
CMP R2,#SREG5+2 ; WAS FDST ADDR REG INCRE RIGHT AMOUNT?
BEQ 67$ ;
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 216(R2),AC3 ; M7-R2

```

```

2559
2560 012612 012701 001174      MOV    #SREG2,R1      ; ADDR(DEST)
2561 012616 174311      STF    AC3,(R1)      ; M1-R1
2562
2563 012620 026767 166344 166346  CMP    $REG0,$REG2   ; 1ST WORD PATTERN CHECK?
2564 012626 001004      BNE    64$           ;
2565 012630 026767 166336 166340  CMP    $REG1,$REG3   ; 2ND WORD PATTERN CHECK?
2566 012636 001401      BEQ    65$           ;
2567 012640 104062      64$:  ERROR    62      ; PATTERN DOESNT MATCH
2568
2569 012642      65$:
2570 012642 000402      BR     TST150        ;;
2571
2572 012644 125252 125252  ADRF4: .WORD  ALTN,ALTN ; TEST PATTERN
2573
2574
2575 ;:*****
2576 ;:TEST 150      TEST OF LOAD-FSRC MODE-3(PC), STORE-FDST MODE-2, F MODE
2577 ;:*****
2578 012650 000004      †TST150: SCOPE
2579 012652 170127 047400  LDFPS  #047400      ; F MODE FPS
2580 012656 016767 000056 166304  MOV    ADRF5+0,$REG0 ; MOVE PATTERN
2581 012664 016767 000052 166300  MOV    ADRF5+2,$REG1 ;
2582
2583 012672 172437 001170      LDF    2*$REG0,AC0   ; M3-R7
2584
2585 012676 012703 001174      MOV    #SREG2,R3     ; ADDR(DEST)
2586 012702 174023      STF    AC0,(R3)+     ; M2-R3
2587
2588 012704 020327 001200      CMP    R3,#SREG2+4   ; WAS FDST ADDR REG INCRE RIGHT AMOUNT?
2589 012710 001401      BEQ    66$           ;
2590 012712 104063      66$:  ERROR    63      ; NOT EQUAL, SIGNAL ERROR
2591 012714
2592
2593 012714 026767 166250 166252  CMP    $REG0,$REG2   ; 1ST WORD PATTERN CHECK?
2594 012722 001004      BNE    64$           ;
2595 012724 026767 166242 166244  CMP    $REG1,$REG3   ; 2ND WORD PATTERN CHECK?
2596 012732 001401      BEQ    65$           ;
2597 012734 104064      64$:  ERROR    64      ; PATTERN DOESNT MATCH
2598
2599 012736      65$:
2600 012736 000402      BR     TST151        ;;
2601
2602 012740 007417 007417  ADRF5: .WORD  ALT4P,ALT4P ; TEST PATTERN
2603
2604
2605 ;:*****
2606 ;:TEST 151      TEST OF LOAD-FSRC MODE-7(PC), STORE-FDST MODE-7, F MODE
2607 ;:*****
2608 012744 000004      †TST151: SCOPE
2609 012746 170127 047400  LDFPS  #047400      ; F MODE FPS
2610 012752 016767 000064 166210  MOV    ADRF6+0,$REG0 ; MOVE PATTERN
2611 012760 016767 000060 166204  MOV    ADRF6+2,$REG1 ;
2612
2613 012766 012767 001170 166204  MOV    #SREG0,$REG4  ; ADDR(DATA)
2614 012774 172577 166200  LDF    2*$REG4,AC1   ; M7-R7

```

```

2615
2616 013000 012767 001174 166174      MOV    #SREG2,SREG5      ; ADDR(DEST)
2617 013006 012705 001226      MOV    #SREG5+24,R5     ; ADDR(ADDR(DEST))+24
2618 013012 174175 177754      STF    AC1,2-24(R5)     ; M7-R5
2619
2620 013016 026767 166146 166150      CMP    $REG0,$REG2      ; 1ST WORD PATTERN CHECK?
2621 013024 001004 64$          BNE    64$              ;
2622 013026 026767 166140 166142      CMP    $REG1,$REG3      ; 2ND WORD PATTERN CHECK?
2623 013034 001401 65$          BEQ    65$              ;
2624 013036 104065 64$:      ERROR  65              ; PATTERN DOESNT MATCH
2625
2626 013040 65$:
2627 013040 000402      BR     TST152           ;;
2628
2629 013042 125252 125252      ADRF6: .WORD  ALTN,ALTN ; TEST PATTERN
2630
2631
2632
2633 ;*****
2633 ;*TEST 152      TEST OF LOAD-FSRC MODE-2, STORE-FDST MODE-3(PC), F MODE
2633 ;*****
2634
2635 013046 000004      †TST152: SCOPE
2636 013050 170127 047400      LDFPS #047400           ; F MODE FPS
2637 013054 016767 000056 166106      MOV    ADRF7+0,$REG0    ; MOVE PATTERN
2638 013062 016767 000052 166102      MOV    ADRF7+2,$REG1    ;
2639
2640 013070 012703 001170      MOV    #SREG0,R3        ; ADDR(DATA)
2641 013074 172623      LDF    (R3)+,AC2        ; M2-R3
2642
2643 013076 174237 001174      STF    AC2,2#SREG2      ; M3-R7
2644
2645 013102 020327 001174      CMP    R3,#SREG0+4      ; WAS FSRC ADDR REG INCRE RIGHT AMOUNT?
2646 013106 001401 66$          BEQ    66$              ;
2647 013110 104066 66$:      ERROR  66              ; NOT EQUAL, SIGNAL ERROR
2648 013112
2649
2650 013112 026767 166052 166054      CMP    $REG0,$REG2      ; 1ST WORD PATTERN CHECK?
2651 013120 001004 64$          BNE    64$              ;
2652 013122 026767 166044 166046      CMP    $REG1,$REG3      ; 2ND WORD PATTERN CHECK?
2653 013130 001401 65$          BEQ    65$              ;
2654 013132 104064 64$:      ERROR  64              ; PATTERN DOESNT MATCH
2655
2656 013134 65$:
2657 013134 000402      BR     TST153           ;;
2658
2659 013136 170360 170360      ADRF7: .WORD  ALT4N,ALT4N ; TEST PATTERN
2660
2661
2662
2663 ;*****
2663 ;*TEST 153      TEST OF LOAD-FSRC MODE-4, STORE-FDST MODE-5, F MODE
2663 ;*****
2664
2665 013142 000004      †TST153: SCOPE
2666 013144 170127 047400      LDFPS #047400           ; F MODE FPS
2667 013150 016767 000076 166012      MOV    ADRF10+0,$REG0   ; MOVE PATTERN
2668 013156 016767 000072 166006      MOV    ADRF10+2,$REG1   ;
2669
2670 013164 012705 001174      MOV    #SREG0+4,R5      ; ADDR(DATA+4)

```

K06

FPU BASIC INSTR TESTS MACY11 27(1006) 25-APR-77 09:12 PAGE 56  
 DQFPAB.P11 19-APR-77 13:36 T153 TEST OF LOAD-FSRC MODE-4, STORE-FDST MODE-5, F MODE

SEQ 0060

2671	013170	172745			LDF	-(R5),AC3		M4-R5
2672								
2673	013172	012767	001174	166000	MOV	#\$REG2,\$REG4		ADDR(DEST)
2674	013200	012704	001202		MOV	#\$REG4+2,R4		ADDR(ADDR(DEST)+2)
2675	013204	174354			STF	AC3,2-(R4)		M5-R4
2676								
2677	013206	020527	001170		CMP	R5,\$REG0		WAS FSRC ADDR REG DECRE RIGHT AMOUNT?
2678	013212	001401			BEQ	66\$		
2679	013214	104067			ERROR	67		NOT EQUAL, SIGNAL ERROR
2680	013216				66\$:			
2681								
2682	013216	020427	001200		CMP	R4,\$REG4		WAS FDST ADDR REG DECRE RIGHT AMOUNT?
2683	013222	001401			BEQ	67\$		
2684	013224	104070			ERROR	70		NOT EQUAL, SIGNAL ERROR
2685	013226				67\$:			
2686								
2687	013226	026767	165736	165740	CMP	\$REG0,\$REG2		1ST WORD PATTERN CHECK?
2688	013234	001004			BNE	64\$		
2689	013236	026767	165730	165732	CMP	\$REG1,\$REG3		2ND WORD PATTERN CHECK?
2690	013244	001401			BEQ	65\$		
2691	013246	104071			ERROR	71		PATTERN DOESNT MATCH
2692					64\$:			
2693	013250				65\$:			
2694	013250	000402			BR	TST154		::
2695								
2696	013252	052525	052525		ADRF10:	.WORD ALTP,ALTP		TEST PATTERN
2697								
2698								
2699								
2700								
2701								
2702	013256	000004			TST154:	SCOPE		
2703	013260	170127	047400		LDFPS	#047400		F MODE FPS
2704	013264	016767	000070	165676	MOV	ADRF11+0,\$REG0		MOVE PATTERN
2705	013272	016767	000064	165672	MOV	ADRF11+2,\$REG1		
2706								
2707	013300	016767	165664	000002	MOV	\$REG0,1\$		PUT DATA
2708	013306	172427			LDF	(PC)+,ACO		M2-R7
2709	013310	000000			.WORD	0		DATA
2710	013312	000403			BR	2\$		THIS SHOULD BE NEXT INSTR AFTER FPP PC MODE 2
2711	013314	104120			ERROR	120		NOT HERE (+4 INCRE)
2712	013316	104121			ERROR	121		OR HERE (+6 INCRE)
2713	013320	104122			ERROR	122		OR HERE (+10 INCRE)
2714								
2715	013322	012767	001174	165650	2\$:	MOV	#\$REG2,\$REG4	ADDR(DEST)
2716	013330	174077	165644		STF	ACO,2\$REG4		M7-R7
2717								
2718	013334	026767	165630	165632	CMP	\$REG0,\$REG2		1ST WORD PATTERN CHECK?
2719	013342	001004			BNE	64\$		
2720	013344	026767	165622	165624	CMP	\$REG1,\$REG3		2ND WORD PATTERN CHECK?
2721	013352	001401			BEQ	65\$		
2722	013354	104072			ERROR	72		PATTERN DOESNT MATCH
2723					64\$:			
2724	013356				65\$:			
2725	013356	000402			BR	TST155		::
2726								

2727 013360 104117 000000

ADRF11: .WORD ERROR117,0 ; TEST PATTERN

2728  
2729

2730

2731

2732

2733 013364 000004

\*\*\*\*\*  
: TEST 155 TEST OF LOAD-FSRC MODE-6(PC), STORE-FDST MODE-6(PC), F MODE  
: \*\*\*\*\*

2734 013366 170127 047400

TST155: SCOPE

2735 013372 016767 000044 165570

LDFPS #047400 ; F MODE FPS

2736 013400 016767 000040 165564

MOV ADRF12+0,\$REG0 ; MOVE PATTERN

2737

MOV ADRF12+2,\$REG1 ;

2738 013406 172567 165556

LDF \$REG0,AC1 ; M6-R7

2739

STF AC1,\$REG2 ; M6-R7

2740 013412 174167 165556

STF AC1,\$REG2 ; M6-R7

2741

CMP \$REG0,\$REG2 ; 1ST WORD PATTERN CHECK?

2742 013416 026767 165546 165550

BNE 64\$ ;

2743 013424 001004

CMP \$REG1,\$REG3 ; 2ND WORD PATTERN CHECK?

2744 013426 026767 165540 165542

BEQ 65\$ ;

2745 013434 001401

64\$: ERROR 73 ; PATTERN DOESNT MATCH

2746 013436 104073

65\$:

2747

BR TST156 ; ;

2748 013440

ADRF12: .WORD ALTP,ALTP ; TEST PATTERN

2749 013440 000402

2750

2751 013442 052525 052525

ADRF12: .WORD ALTP,ALTP ; TEST PATTERN

2752

2753

2754

2755

2756

2757 013446 000004

\*\*\*\*\*  
: TEST 156 TEST OF LOAD-FSRC MODE-0, STORE-FDST MODE-2(PC), F MODE  
: \*\*\*\*\*

2758 013450 170127 047400

TST156: SCOPE

2759 013454 016767 000062 165506

LDFPS #047400 ; F MODE FPS

2760 013462 016767 000056 165502

MOV ADRF13+0,\$REG0 ; MOVE PATTERN

2761

MOV ADRF13+2,\$REG1 ;

2762 013470 172667 165474

LDF \$REG0,AC2 ; M6-R7 WORKS

2763 013474 172702

LDF AC2,AC3 ; M0-R2

2764

MOV #ERROR117,1\$ ; SETUP ERROR CALL FOR WRONG INCREMENT

2765 013476 012767 104117 000002

STF AC3,(PC)+ ; M2-R7

2766 013504 174327

1\$: .WORD 0 ; DEST

2767 013506 000000

BR 2\$ ; THIS SHOULD BE NEXT INSTR AFTER FPP PC MODE 2

2768 013510 000403

ERROR 120 ; NOT HERE (+4 INCRE)

2769 013512 104120

ERROR 121 ; OR HERE (+6 INCRE)

2770 013514 104121

ERROR 122 ; OR HERE (+10 INCRE)

2771 013516 104122

2\$: MOV 1\$, \$REG2 ; GET DEST

2772 013520 016767 177762 165446

CMP \$REG0,\$REG2 ; WAS 1 WORD STORED OK?

2773

BEQ 64\$ ;

2774 013526 026767 165436 165440

64\$: ERROR 73 ; NOT EQUAL, SIGNAL ERROR

2775 013534 001401

BR TST157 ; ;

2776 013536 104073

ADRF13: .WORD ALT4P,0 ; TEST PATTERN

2777 013540

2778

2779 013540 000402

2780

2781 013542 007417 000000

2782

M06

FPU BASIC INSTR TESTS MACY11 27(1006) 25-APR-77 09:12 PAGE 58  
DQFPAB.P11 19-APR-77 13:36

T156 TEST OF LOAD-FSRC MODE-0, STORE-FDST MODE-2(PC), F MODE

SEQ 0062

```

2783
2784
2785
2786
2787 013546 000004
2788 013550 170127 047400
2789 013554 016767 000052 165406
2790 013562 016767 000046 165402
2791
2792 013570 012701 001200
2793 013574 172461 177770
2794
2795 013600 174001
2796 013602 174167 165366
2797
2798 013606 026767 165356 165360
2799 013614 001004
2800 013616 026767 165350 165352
2801 013624 001401
2802 013626 104074
2803
2804 013630
2805 013630 000402
2806
2807 013632 007417 007417
2808
2809

```

```

*****
*TEST 157 TEST OF LOAD-FSRC MODE-6, STORE-FDST MODE-0, F MODE
*****
TST157: SCOPE
LDFPS 8047400 ; F MODE FPS
MOV ADRF14+0,$REG0 ; MOVE PATTERN
MOV ADRF14+2,$REG1 ;
MOV #$REG0+10,R1 ; ADDR(DATA)+10
LDF -10(R1),AC0 ; M6-R1
STF AC0,AC1 ; M0-R1
STF AC1,$REG2 ; M6-R7 WORKS
CMP $REG0,$REG2 ; 1ST WORD PATTERN CHECK?
BNE 64$ ;
CMP $REG1,$REG3 ; 2ND WORD PATTERN CHECK?
BEQ 65$ ;
64$: ERROR 74 ; PATTERN DOESNT MATCH
65$:
BR TST160 ;;
ADRF14: .WORD ALT4P,ALT4P ; TEST PATTERN

```

```

2810
2811
2812
2813
2814
2815
2816
2817
2818 013636 000004
2819 013640 170127 047400
2820 013644 005037 001170
2821 013650 012737 000005 001172
2822 013656 172437 001170
2823 013662 174005
2824 013664 005337 001172
2825 013670 172437 001170
2826 013674 174004
2827 013676 005337 001172
2828 013702 172737 001170
2829 013706 005337 001172
2830 013712 172637 001170
2831 013716 005337 001172
2832 013722 172537 001170
2833 013726 005337 001172
2834 013732 172437 001170
2835 013736 005037 001170
2836 013742 005037 001172
2837 013746 174037 001174
2838 013752 005737 001174
2839 013756 001004
2840 013760 023737 001172 001176
2841 013766 001401
2842 013770 104075 1$:
2843 013772 005237 001172 11$:
2844 013776 174137 001174
2845 014002 005737 001174
2846 014006 001004
2847 014010 023737 001172 001176
2848 014016 001401
2849 014020 104075 2$:
2850 014022 005237 001172 21$:
2851 014026 174237 001174
2852 014032 005737 001174
2853 014036 001004
2854 014040 023737 001172 001176
2855 014046 001401
2856 014050 104075 3$:
2857 014052 005237 001172 31$:
2858 014056 174337 001174
2859 014062 005737 001174
2860 014066 001004
2861 014070 023737 001172 001176
2862 014076 001401
2863 014100 104075 4$:
2864 014102 005237 001172 41$:
2865 014106 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 SREG0 ; SET AC0 = 0,
MOV #5,SREG1 ; AC1 = 1,
LDF SREG0,AC0 ; AC2 = 2,
STF AC0,AC5 ; AC3 = 3,
DEC SREG1 ; AC4 = 4,
LDF SREG0,AC0 ; AC5 = 5
STF AC0,AC4
DEC SREG1
LDF SREG0,AC3
DEC SREG1
LDF SREG0,AC2
DEC SREG1
LDF SREG0,AC1
DEC SREG1
LDF SREG0,AC0
CLR SREG0
CLR SREG1
STF AC0,SREG2
TST SREG2
BNE 1$
CMP SREG1,SREG3
BEQ 11$
1$: ERROR 7$
11$: INC SREG1
STF AC1,SREG2
TST SREG2
BNE 2$
CMP SREG1,SREG3
BEQ 21$
2$: ERROR 7$
21$: INC SREG1
STF AC2,SREG2
TST SREG2
BNE 3$
CMP SREG1,SREG3
BEQ 31$
3$: ERROR 7$
31$: INC SREG1
STF AC3,SREG2
TST SREG2
BNE 4$
CMP SREG1,SREG3
BEQ 41$
4$: ERROR 7$
41$: INC SREG1
LDF AC4,AC0
NOW TEST THE ABOVE IS TRUE:
DOES AC0 = 0 ?
...
NO, ERROR
YES, INC FOR NEXT TEST
DOES AC1 = 1 ?
...
NO, ERROR
YES, INC FOR NEXT TEST
DOES AC2 = 2 ?
...
NO, ERROR
YES, INC FOR NEXT TEST
DOES AC3 = 3 ?
...
NO, ERROR
YES, INC FOR NEXT TEST
DOES AC4 = 4 ?

```

2866	014110	174037	001174		STF	AC0, \$REG2	:	...
2867	014114	005737	001174		TST	\$REG2	:	...
2868	014120	001004			BNE	5\$	:	...
2869	014122	023737	001172	001176	CMP	\$REG1, \$REG3	:	...
2870	014130	001401			BEQ	51\$	:	...
2871	014132	104075			ERROR	7\$	:	NO, ERROR
2872	014134	005237	001172		INC	\$REG1	:	YES, INC FOR NEXT TEST
2873	014140	172405			LDF	AC5, AC0	:	DOES AC5 = 5 ?
2874	014142	174037	001174		STF	AC0, \$REG2	:	...
2875	014146	005737	001174		TST	\$REG2	:	...
2876	014152	001004			BNE	6\$	:	...
2877	014154	023737	001172	001176	CMP	\$REG1, \$REG3	:	...
2878	014162	001401			BEQ	61\$	:	...
2879	014164	104075			ERROR	7\$	:	NO, ERROR
2880	014166						:	YES, GO FOR NEXT TEST



```

2881
2882
2883
2884
2885 014166 000004
2886 014170 012705 014202
2887 014174 004737 021704
2888
2889 014200 000407
2890
2891 014202
2892 014202 000000 000000
2893 014206 000000 000000
2894 014212 047453 047444
2895 014216 000000
2896
2897
2898
2899
2900
2901
2902 014220 000004
2903 014222 012705 014234
2904 014226 004737 021704
2905
2906 014232 000407
2907
2908 014234
2909 014234 052525 052525
2910 014240 052525 052525
2911 014244 047557 047540
2912 014250 000000
2913
2914
2915
2916
2917
2918
2919 014252 000004
2920 014254 012705 014266
2921 014260 004737 021704
2922
2923 014264 000407
2924
2925 014266
2926 014266 125252 125252
2927 014272 025252 125252
2928 014276 047417 047400
2929 014302 000000
2930
2931
2932
2933
2934
2935
2936 014304 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

```

2937 014306 012705 014320  
 2938 014312 004737 021704  
 2939  
 2940 014316 000407  
 2941  
 2942 014320  
 2943 014320 177777 177777  
 2944 014324 077777 177777  
 2945 014330 047517 047500  
 2946 014334 000000  
 2947  
 2948  
 2949  
 2950  
 2951  
 2952  
 2953 014336 000004  
 2954 014340 012705 014352  
 2955 014344 004737 021704  
 2956  
 2957 014350 000407  
 2958  
 2959 014352  
 2960 014352 000200 000000  
 2961 014356 000200 000000  
 2962 014362 047457 047440  
 2963 014366 000000  
 2964  
 2965  
 2966  
 2967  
 2968  
 2969  
 2970 014370 000004  
 2971 014372 012705 014404  
 2972 014376 004737 021704  
 2973  
 2974 014402 000407  
 2975  
 2976 014404  
 2977 014404 077777 177777  
 2978 014410 077777 177777  
 2979 014414 047557 047540  
 2980 014420 000000  
 2981  
 2982  
 2983  
 2984  
 2985  
 2986  
 2987 014422 000004  
 2988 014424 012705 014436  
 2989 014430 004737 021704  
 2990  
 2991 014434 000407  
 2992

```

MOV #ABSF4,R5 ; PTR TO TEST DATA SET
JSR PC,@#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,@#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,@#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,@#ABSFT ; GO TEST

BR TST170 ;;

```

```

2993 014436
2994 014436 100200 000000
2995 014442 000200 000000
2996 014446 047417 047400
2997 014452 000000
2998
2999
3000
3001
3002
3003
3004 014454 000004
3005 014456 012705 014470
3006 014462 004737 021704
3007
3008 014466 000407
3009
3010 014470
3011 014470 000177 177777
3012 014474 000000 000000
3013 014500 047513 047504
3014 014504 000000
3015
3016
3017
3018
3019
3020
3021 014506 000004
3022 014510 012705 014522
3023 014514 004737 021704
3024
3025 014520 000407
3026
3027 014522
3028 014522 100000 000000
3029 014526 000000 000000
3030 014532 047453 147444
3031 014536 100014
3032
3033
3034
3035
3036
3037
3038 014540 000004
3039 014542 012705 014554
3040 014546 004737 021704
3041
3042 014552 000407
3043
3044 014554
3045 014554 100000 000001
3046 014560 000000 000000
3047 014564 047553 147544
3048 014570 100014

```

```

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 )

```

```

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

```

```

TST170: SCOPE
MOV #ABSF10_RS ; PTR TO TEST DATA SET
JSR PC, @ABSF10 ; GO TEST
BR TST171 ;;

```

```

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 )

```

```

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

```

```

TST171: SCOPE
MOV #ABSF11_RS ; PTR TO TEST DATA SET
JSR PC, @ABSF11 ; GO TEST
BR TST172 ;;

```

```

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 )

```

```

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

```

```

TST172: SCOPE
MOV #ABSF12_RS ; PTR TO TEST DATA SET
JSR PC, @ABSF12 ; GO TEST
BR TST173 ;;

```

```

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 )

```

```

3049
3050
3051
3052
3053
3054
3055 014572 000004
3056 014574 012705 014606
3057 014600 004737 021704
3058
3059 014604 000407
3060
3061 014606
3062 014606 100177 177777
3063 014612 000000 000000
3064 014616 043413 043404
3065 014622 000000
3066
3067
3068

```

```

*****
*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,@ABSF ; 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 )

```

```

3069
3070
3071
3072
3073 014624 000004
3074 014626 012705 014640
3075 014632 004737 022060
3076
3077 014636 000413
3078
3079 014640
3080 014640 000000 000000 000000
3081 014646 000000
3082 014650 000000 000000 000000
3083 014656 000000
3084 014660 047713 047704
3085 014664 000000
3086
3087
3088
3089
3090
3091
3092 014666 000004
3093 014670 012705 014702
3094 014674 004737 022060
3095
3096 014700 000413
3097
3098 014702
3099 014702 052525 052525 052525
3100 014710 052525
3101 014712 052525 052525 052525
3102 014720 052525
3103 014722 047657 047640
3104 014726 000000
3105
3106
3107
3108
3109
3110
3111 014730 000004
3112 014732 012705 014744
3113 014736 004737 022060
3114
3115 014742 000413
3116
3117 014744
3118 014744 125252 125252 125252
3119 014752 125252
3120 014754 025252 125252 125252
3121 014762 125252
3122 014764 047757 047740
3123 014770 000000
3124

```

```

*****
; 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 )

```

3125  
 3126  
 3127  
 3128  
 3129  
 3130 014772 000004  
 3131 014774 012705 015006  
 3132 015000 004737 022060  
 3133  
 3134 015004 000413  
 3135  
 3136 015006  
 3137 015006 177777 177777 177777  
 3138 015014 177777  
 3139 015016 077777 177777 177777  
 3140 015024 177777  
 3141 015026 047617 047600  
 3142 015032 000000  
 3143  
 3144  
 3145  
 3146  
 3147  
 3148  
 3149 015034 000004  
 3150 015036 012705 015050  
 3151 015042 004737 022060  
 3152  
 3153 015046 000413  
 3154  
 3155 015050  
 3156 015050 000200 000000 000000  
 3157 015056 000000  
 3158 015060 000200 000000 000000  
 3159 015066 000000  
 3160 015070 047717 047700  
 3161 015074 000000  
 3162  
 3163  
 3164  
 3165  
 3166  
 3167  
 3168 015076 000004  
 3169 015100 012705 015112  
 3170 015104 004737 022060  
 3171  
 3172 015110 000413  
 3173  
 3174 015112  
 3175 015112 077777 177777 177777  
 3176 015120 177777  
 3177 015122 077777 177777 177777  
 3178 015130 177777  
 3179 015132 047657 047640  
 3180 015136 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 )

```

```

3181
3182
3183
3184
3185
3186
3187 015140 000004
3188 015142 012705 015154
3189 015146 004737 022060
3190
3191 015152 000413
3192
3193 015154
3194 015154 100200 000000 000000
3195 015162 000000
3196 015164 000200 000000 000000
3197 015172 000000
3198 015174 047757 047740
3199 015200 000000
3200
3201
3202
3203
3204
3205
3206 015202 000004
3207 015204 012705 015216
3208 015210 004737 022060
3209
3210 015214 000413
3211
3212 015216
3213 015216 000177 177777 177777
3214 015224 177777
3215 015226 000000 000000 000000
3216 015234 000000
3217 015236 047613 047604
3218 015242 000000
3219
3220
3221
3222
3223
3224
3225 015244 000004
3226 015246 012705 015260
3227 015252 004737 022060
3228
3229 015256 000413
3230
3231 015260
3232 015260 100000 000000 000000
3233 015266 000000
3234 015270 000000 000000 000000
3235 015276 000000
3236 015300 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 MO,0,0,0 ; INITIAL MEM FLOAT NUMBER
.WORD 0,0,0,0 ; EXPECTED FLOAT RESULT
.WORD 047713,147704 ; FPS: BEFORE, AFTER

```

3237 015304 100014 .WORD 100014 ; FEC AFTER ( 0 = N/A )

3238  
3239  
3240  
3241  
3242  
3243  
3244  
3245  
3246  
3247  
3248  
3249  
3250  
3251  
3252  
3253  
3254  
3255  
3256  
3257  
3258  
3259  
3260  
3261  
3262  
3263  
3264  
3265  
3266  
3267  
3268  
3269  
3270  
3271  
3272  
3273  
3274  
3275  
3276  
3277  
3278

\*\*\*\*\*  
\*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 )



```

3279
3280
3281
3282
3283 015412 000004
3284 015414 012705 015426
3285 015420 004737 022270
3286
3287 015424 000407
3288
3289 015426
3290 015426 000000 000000
3291 015432 000000 000000
3292 015436 047413 047404
3293 015442 000000
3294
3295
3296
3297
3298
3299
3300 015444 000004
3301 015446 012705 015460
3302 015452 004737 022270
3303
3304 015456 000407
3305
3306 015460
3307 015460 052525 052525
3308 015464 152525 052525
3309 015470 047547 047550
3310 015474 000000
3311
3312
3313
3314
3315
3316
3317 015476 000004
3318 015500 012705 015512
3319 015504 004737 022270
3320
3321 015510 000407
3322
3323 015512
3324 015512 125252 125252
3325 015516 025252 125252
3326 015522 047517 047500
3327 015526 000000
3328
3329
3330
3331
3332
3333
3334 015530 000004

```

```

*****
*TEST 207 TEST OF NEGF INSTR, DATA SET NEGF-1
* ROUNDING MODE, ALL INTERRUPT ENABLES ON
*****
TST207: SCOPE
MOV #NEGF1,R5 ; PTR TO TEST DATA SET
JSR PC,@#NEGF1 ; 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,@#NEGF2 ; 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
* ROUNDING MODE, ALL INTERRUPT ENABLES ON
*****
TST211: SCOPE
MOV #NEGF3,R5 ; PTR TO TEST DATA SET
JSR PC,@#NEGF3 ; 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

```

3335 015532 012705 015544  
 3336 015536 004737 022270  
 3337  
 3338 015542 000407  
 3339  
 3340 015544  
 3341 015544 177777 177777  
 3342 015550 077777 177777  
 3343 015554 047457 047440  
 3344 015560 000000  
 3345  
 3346  
 3347  
 3348  
 3349  
 3350  
 3351 015562 000004  
 3352 015564 012705 015576  
 3353 015570 004737 022270  
 3354  
 3355 015574 000407  
 3356  
 3357 015576  
 3358 015576 000200 000000  
 3359 015602 100200 000000  
 3360 015606 047407 047410  
 3361 015612 000000  
 3362  
 3363  
 3364  
 3365  
 3366  
 3367  
 3368 015614 000004  
 3369 015616 012705 015630  
 3370 015622 004737 022270  
 3371  
 3372 015626 000407  
 3373  
 3374 015630  
 3375 015630 077777 177777  
 3376 015634 177777 177777  
 3377 015640 047547 047550  
 3378 015644 000000  
 3379  
 3380  
 3381  
 3382  
 3383  
 3384  
 3385 015646 000004  
 3386 015650 012705 015662  
 3387 015654 004737 022270  
 3388  
 3389 015660 000407  
 3390

```

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

BR TST213 ;;

NEGF4: ; TEST DATA SET NEGF-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 NEGF INSTR, DATA SET NEGF-5
* ROUNDING MODE, ALL INTERRUPT ENABLES ON
*****

```

```

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

BR TST214 ;;

```

```

NEGF5: ; TEST DATA SET NEGF-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 NEGF INSTR, DATA SET NEGF-6
* TRUNCATE MODE, ALL INTERRUPT ENABLES ON
*****

```

```

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

BR TST215 ;;

```

```

NEGF6: ; TEST DATA SET NEGF-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 NEGF INSTR, DATA SET NEGF-7
* ROUNDING MODE, ALL INTERRUPT ENABLES ON
*****

```

```

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

BR TST216 ;;

```

T215 TEST OF NEGF INSTR, DATA SET NEGF-7

3391	015662		
3392	015662	100200	000000
3393	015666	000200	000000
3394	015672	047517	047500
3395	015676	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 )

```

3396  
3397  
3398

```

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

```

3401			
3402	015700	000004	
3403	015702	012705	015714
3404	015706	004737	022270
3405			

```

TST216: SCOPE
        MOV     #NEGF10,R5 ; PTR TO TEST DATA SET
        JSR    PC,#NEGF7 ; GO TEST
        BR     TST217 ;

```

3406  
3407

```

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 )

```

3408	015714		
3409	015714	000177	177777
3410	015720	000000	000000
3411	015724	047453	047444
3412	015730	000000	

3413  
3414  
3415

```

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

```

3416			
3417			
3418			
3419	015732	000004	
3420	015734	012705	015746
3421	015740	004737	022270

```

TST217: SCOPE
        MOV     #NEGF11,R5 ; PTR TO TEST DATA SET
        JSR    PC,#NEGF7 ; GO TEST
        BR     TST220 ;

```

3422  
3423  
3424

```

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 )

```

3425	015746		
3426	015746	100000	000000
3427	015752	000000	000000
3428	015756	047453	147444
3429	015762	100014	

3430  
3431  
3432

```

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

```

3433			
3434			
3435			
3436	015764	000004	
3437	015766	012705	016000
3438	015772	004737	022270

```

TST220: SCOPE
        MOV     #NEGF12,R5 ; PTR TO TEST DATA SET
        JSR    PC,#NEGF7 ; GO TEST
        BR     TST221 ;

```

3439  
3440  
3441

```

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 )

```

3442	016000		
3443	016000	100000	000001
3444	016004	000000	000000
3445	016010	047553	147544
3446	016014	100014	

3447  
3448  
3449  
3450  
3451  
3452  
3453  
3454  
3455  
3456  
3457  
3458  
3459  
3460  
3461  
3462  
3463  
3464  
3465  
3466

016016	000004		
016020	012705	016032	
016024	004737	022270	
016030	000407		
016032			
016032	100177	177777	
016036	000000	000000	
016042	043513	043504	
016046	000000		

```

*****
:TEST 221      TEST OF NEGF INSTR, DATA SET NEGF-13
:*            ROUNGING 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,M!       ; INITIAL MEM FLOAT NUMBER
.WORD       0,0           ; EXPECTED FLOAT RESULT
.WORD       043513,043504 ; FPS: BEFORE, AFTER
.WORD       000000       ; FEC AFTER ( 0 = N/A )

```

```

3467
3468
3469
3470
3471 016050 000004
3472 016052 012705 016064
3473 016056 004737 022444
3474
3475 016062 000413
3476
3477 016064
3478 016064 000000 000000 000000
3479 016072 000000
3480 016074 000000 000000 000000
3481 016102 000000
3482 016104 047753 047744
3483 016110 000000
3484
3485
3486
3487
3488
3489
3490 016112 000004
3491 016114 012705 016126
3492 016120 004737 022444
3493
3494 016124 000413
3495
3496 016126
3497 016126 052525 052525 052525
3498 016134 052525
3499 016136 152525 052525 052525
3500 016144 052525
3501 016146 047607 047610
3502 016152 000000
3503
3504
3505
3506
3507
3508
3509 016154 000004
3510 016156 012705 016170
3511 016162 004737 022444
3512
3513 016166 000413
3514
3515 016170
3516 016170 125252 125252 125252
3517 016176 125252
3518 016200 025252 125252 125252
3519 016206 125252
3520 016210 047657 047640
3521 016214 000000
3522

```

```

*****
*TEST 222 TEST OF NEG0 INSTR, DATA SET NEG0-1
* TRUNCATE MODE, ALL INTERRUPT ENABLES ON
*****
TST222: SCOPE
MOV #NEG01,R5 ; PTR TO TEST DATA SET
JSR PC,@#NEG0T ; GO TEST
BR TST223 ;;
NEG01: ; TEST DATA SET NEG0-1:
.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 NEG0 INSTR, DATA SET NEG0-2
* ROUNDING MODE, ALL INTERRUPT ENABLES ON
*****
TST223: SCOPE
MOV #NEG02,R5 ; PTR TO TEST DATA SET
JSR PC,@#NEG0T ; GO TEST
BR TST224 ;;
NEG02: ; TEST DATA SET NEG0-2:
.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 NEG0 INSTR, DATA SET NEG0-3
* TRUNCATE MODE, ALL INTERRUPT ENABLES ON
*****
TST224: SCOPE
MOV #NEG03,R5 ; PTR TO TEST DATA SET
JSR PC,@#NEG0T ; GO TEST
BR TST225 ;;
NEG03: ; TEST DATA SET NEG0-3:
.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 )

```

```

3523
3524
3525
3526
3527
3528 016216 000004
3529 016220 012705 016232
3530 016224 004737 022444
3531
3532 016230 000413
3533
3534 016232
3535 016232 177777 177777 177777
3536 016240 177777
3537 016242 077777 177777 177777
3538 016250 177777
3539 016252 047717 047700
3540 016256 000000
3541
3542
3543
3544
3545
3546
3547 016260 000004
3548 016262 012705 016274
3549 016266 004737 022444
3550
3551 016272 000413
3552
3553 016274
3554 016274 000200 000000 000000
3555 016302 000000
3556 016304 100200 000000 000000
3557 016312 000000
3558 016314 047747 047750
3559 016320 000000
3560
3561
3562
3563
3564
3565
3566 016322 000004
3567 016324 012705 016336
3568 016330 004737 022444
3569
3570 016334 000413
3571
3572 016336
3573 016336 077777 177777 177777
3574 016344 177777
3575 016346 177777 177777 177777
3576 016354 177777
3577 016356 047607 047610
3578 016362 000000

```

```

*****
;TEST 225 TEST OF NEG0 INSTR, DATA SET NEG0-4
;* ROUNING MODE, ALL INTERRUPT ENABLES ON
*****

```

```

†ST225: SCOPE
MOV #NEG04,RS ; PTR TO TEST DATA SET
JSR PC,‡NEG0T ; GO TEST
BR TST226 ;;
NEG04: ; TEST DATA SET NEG0-4:
.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 NEG0 INSTR, DATA SET NEG0-5
;* TRUNCATE MODE, ALL INTERRUPT ENABLES ON
*****

```

```

†ST226: SCOPE
MOV #NEG05,RS ; PTR TO TEST DATA SET
JSR PC,‡NEG0T ; GO TEST
BR TST227 ;;
NEG05: ; TEST DATA SET NEG0-5:
.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 NEG0 INSTR, DATA SET NEG0-6
;* ROUNING MODE, ALL INTERRUPT ENABLES ON
*****

```

```

†ST227: SCOPE
MOV #NEG06,RS ; PTR TO TEST DATA SET
JSR PC,‡NEG0T ; GO TEST
BR TST230 ;;
NEG06: ; TEST DATA SET NEG0-6:
.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 )

```

```

3579
3580
3581
3582
3583
3584
3585 016364 000004
3586 016366 012705 016400
3587 016372 004737 022444
3588
3589 016376 000413
3590
3591 016400
3592 016400 100200 000000 000000
3593 016406 000000
3594 016410 000200 000000 000000
3595 016416 000000
3596 016420 047657 047640
3597 016424 000000
3598
3599
3600
3601
3602
3603
3604 016426 000004
3605 016430 012705 016442
3606 016434 004737 022444
3607
3608 016440 000413
3609
3610 016442
3611 016442 000177 177777 177777
3612 016450 177777
3613 016452 000000 000000 000000
3614 016460 000000
3615 016462 047713 047704
3616 016466 000000
3617
3618
3619
3620
3621
3622
3623 016470 000004
3624 016472 012705 016504
3625 016476 004737 022444
3626
3627 016502 000413
3628
3629 016504
3630 016504 100000 000000 000000
3631 016512 000000
3632 016514 000000 000000 000000
3633 016522 000000
3634 016524 047713 147704

```

```

*****
*TEST 230 TEST OF NEG0 INSTR, DATA SET NEG0-7
* TRUNCATE MODE, ALL INTERRUPT ENABLES ON
*****
TST230: SCOPE
MOV #NEG07,R5 ; PTR TO TEST DATA SET
JSR PC,@#NEG0T ; GO TEST
BR TST231 ;;

NEG07: ; TEST DATA SET NEG0-7:
.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 NEG0 INSTR, DATA SET NEG0-10
* ROUNDING MODE, ALL INTERRUPT ENABLES ON
*****
TST231: SCOPE
MOV #NEG010,R5 ; PTR TO TEST DATA SET
JSR PC,@#NEG0T ; GO TEST
BR TST232 ;;

NEG010: ; TEST DATA SET NEG0-10:
.WORD ZXIMP,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 NEG0 INSTR, DATA SET NEG0-11
* ROUNDING MODE, ALL INTERRUPT ENABLES ON
*****
TST232: SCOPE
MOV #NEG011,R5 ; PTR TO TEST DATA SET
JSR PC,@#NEG0T ; GO TEST
BR TST233 ;;

NEG011: ; TEST DATA SET NEG0-11:
.WORD MO,0,0,0 ; INITIAL MEM FLOAT NUMBER
.WORD 0,0,0,0 ; EXPECTED FLOAT RESULT
.WORD 047713,147704 ; FPS: BEFORE, AFTER

```

E08

FPU BASIC INSTR TESTS MACY11 27(1006) 25-APR-77 09:12 PAGE 76  
DQFPAB.P11 19-APR-77 13:36

T232 TEST OF NEGD INSTR, DATA SET NEGD-11

SEQ 0080

3635 016530 100014

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

3636

3637

3638

3639

3640

3641

3642 016532 000004

3643 016534 012705 016546

3644 016540 004737 022444

3645

3646 016544 000413

3647

3648 016546 100000 000000 000000

3649 016546 000001 000000 000000

3650 016554 000000 000000 000000

3651 016556 000000 000000 000000

3652 016564 000000 000000 000000

3653 016566 047613 147604

3654 016572 100014

3655

3656

3657

3658

3659

3660

3661 016574 000004

3662 016576 012705 016610

3663 016602 004737 022444

3664

3665 016606 000413

3666

3667 016610 100177 177777 177777

3668 016610 177777 177777 177777

3669 016616 000000 000000 000000

3670 016620 000000 000000 000000

3671 016626 000000 000000 000000

3672 016630 043753 043744

3673 016634 000000

3674

3675

3676

\*\*\*\*\*

TEST 233 TEST OF NEGD INSTR, DATA SET NEGD-12

\* ROUNDING MODE, ALL INTERRUPT ENABLES ON

\*\*\*\*\*

TST233: SCOPE

MOV #NEGD12,R5 ; PTR TO TEST DATA SET

JSR PC,@#NEGDT ; GO TEST

BR TST234 ;;

NEGD12: ; TEST DATA SET NEGD-12:

.WORD MO,0,0,1 ; INITIAL MEM FLOAT NUMBER

.WORD 0,0,0,0 ; EXPECTED FLOAT RESULT

.WORD 0,0,0,0 ; EXPECTED FLOAT RESULT

.WORD 047613,147604 ; FPS: BEFORE, AFTER

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

\*\*\*\*\*

TEST 234 TEST OF NEGD INSTR, DATA SET NEGD-13

\* TRUNCATE MODE, -0 INTERRUPT ENABLE OFF, ALL OTHERS ON

\*\*\*\*\*

TST234: SCOPE

MOV #NEGD13,R5 ; PTR TO TEST DATA SET

JSR PC,@#NEGDT ; GO TEST

BR TST235 ;;

NEGD13: ; TEST DATA SET NEGD-13:

.WORD ZX1MN,M1,M1,M1 ; INITIAL MEM FLOAT NUMBER

.WORD 0,0,0,0 ; EXPECTED FLOAT RESULT

.WORD 0,0,0,0 ; EXPECTED FLOAT RESULT

.WORD 043753,043744 ; FPS: BEFORE, AFTER

.WORD 000000 ; FEC AFTER ( 0 = N/A )



```

3677
3678
3679
3680
3681 016636 000004
3682 016640 012705 016652
3683 016644 004737 022654
3684
3685 016650 000407
3686
3687 016652
3688 016652 000000 000000
3689 016656 000000 000000
3690 016662 047553 047544
3691 016666 000000
3692
3693
3694
3695
3696
3697
3698 016670 000004
3699 016672 012705 016704
3700 016676 004737 022654
3701
3702 016702 000407
3703
3704 016704
3705 016704 177777 177777
3706 016710 000000 000000
3707 016714 047413 047404
3708 016720 000000
3709
3710
3711
3712
3713
3714
3715 016722 000004
3716 016724 012705 016736
3717 016730 004737 022654
3718
3719 016734 000407
3720
3721 016736
3722 016736 052525 052525
3723 016742 000000 000000
3724 016746 047453 047444
3725 016752 000000
3726
3727
3728
3729
3730
3731
3732 016754 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

```

```

3733 016756 012705 016770
3734 016762 004737 022654
3735
3736 016766 000407
3737
3738 016770
3739 016770 125252 125252
3740 016774 000000 000000
3741 017000 047513 047504
3742 017004 000000
3743
3744
3745
3746
3747
3748
3749 017006 000004
3750 017010 012705 017022
3751 017014 004737 022654
3752
3753 017020 000407
3754
3755 017022
3756 017022 100000 000000
3757 017026 000000 000000
3758 017032 047553 047544
3759 017036 000000
3760
3761
3762
3763
3764
3765
3766 017040 000004
3767 017042 012705 017054
3768 017046 004737 022654
3769
3770 017052 000407
3771
3772 017054
3773 017054 000177 177777
3774 017060 000000 000000
3775 017064 047413 047404
3776 017070 000000
3777
3778
3779

```

```

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 M0,0 ; 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 ZX1MP,M1 ; INITIAL MEM FLOAT NUMBER
.WORD 0,0 ; EXPECTED FLOAT RESULT
.WORD 047413,047404 ; FPS: BEFORE, AFTER
.WORD 000000 ; FEC AFTER ( 0 = N/A )

```

```

3780
3781
3782
3783
3784 017072 000004
3785 017074 012705 017106
3786 017100 004737 023030
3787
3788 017104 000413
3789
3790 017106
3791 017106 000000 000000 000000
3792 017114 000000
3793 017116 000000 000000 000000
3794 017124 000000
3795 017126 047613 047604
3796 017132 000000
3797
3798
3799
3800
3801
3802
3803 017134 000004
3804 017136 012705 017150
3805 017142 004737 023030
3806
3807 017146 000413
3808
3809 017150
3810 017150 177777 177777 177777
3811 017156 177777
3812 017160 000000 000000 000000
3813 017166 000000
3814 017170 047753 047744
3815 017174 000000
3816
3817
3818
3819
3820
3821
3822 017176 000004
3823 017200 012705 017212
3824 017204 004737 023030
3825
3826 017210 000413
3827
3828 017212
3829 017212 052525 052525 052525
3830 017220 052525
3831 017222 000000 000000 000000
3832 017230 000000
3833 017232 047653 047644
3834 017236 000000
3835

```

```

*****
*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 )

```

```

3836
3837
3838
3839
3840
3841 017240 000004
3842 017242 012705 017254
3843 017246 004737 023030
3844
3845 017252 000413
3846
3847 017254
3848 017254 125252 125252 125252
3849 017262 125252
3850 017264 000000 000000 000000
3851 017272 000000
3852 017274 047713 047704
3853 017300 000000
3854
3855
3856
3857
3858
3859
3860 017302 000004
3861 017304 012705 017316
3862 017310 004737 023030
3863
3864 017314 000413
3865
3866 017316
3867 017316 100000 000000 000000
3868 017324 000000
3869 017326 000000 000000 000000
3870 017334 000000
3871 017336 047613 047604
3872 017342 000000
3873
3874
3875
3876
3877
3878
3879 017344 000004
3880 017346 012705 017360
3881 017352 004737 023030
3882
3883 017356 000413
3884
3885 017360
3886 017360 000177 177777 177777
3887 017366 177777
3888 017370 000000 000000 000000
3889 017376 000000
3890 017400 047753 047744
3891 017404 000000

```

```

*****
*TEST 246 TEST OF CLRD INSTR, DATA SET CLRD-4
* ROUNDING MODE, ALL INTERRUPT ENABLES ON
*****
TST246: 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
* ROUNDING MODE, ALL INTERRUPT ENABLES ON
*****
TST247: 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
*****
TST250: SCOPE
MOV #CLRD6,R5 ; PTR TO TEST DATA SET
JSR PC,@#CLRDT ; GO TEST
BR TST251 ;;

CLRD6: ; TEST DATA SET CLRD-6:
.WORD ZX1MP,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 )

```

J08

FPU BASIC INSTR TESTS MACY11 27(1006) 25-APR-77 09:12 PAGE 81  
DQFPAB.P11 19-APR-77 13:36 T250 TEST OF CLRD INSTR, DATA SET CLRD-6

SEQ 0085

3892  
3893  
3894

```

3895
3896
3897
3898
3899 017406 000004
3900 017410 012705 017422
3901 017414 004737 023240
3902
3903 017420 000407
3904
3905 017422
3906 017422 000000 000000
3907 017426 000000 000000
3908 017432 047513 047504
3909 017436 000000
3910
3911
3912
3913
3914
3915
3916 017440 000004
3917 017442 012705 017454
3918 017446 004737 023240
3919
3920 017452 000407
3921
3922 017454
3923 017454 052525 052525
3924 017460 052525 052525
3925 017464 047557 047540
3926 017470 000000
3927
3928
3929
3930
3931
3932
3933 017472 000004
3934 017474 012705 017506
3935 017500 004737 023240
3936
3937 017504 000407
3938
3939 017506
3940 017506 125252 125252
3941 017512 125252 125252
3942 017516 047407 047410
3943 017522 000000
3944
3945
3946
3947
3948
3949
3950 017524 000004

```

```

*****
*TEST 251 TEST OF TSTF INSTR, DATA SET TSTF-1
* ROUNING 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
* ROUNING 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

```

3951 017526 012705 017540  
3952 017532 004737 023240  
3953  
3954 017536 000407  
3955  
3956 017540  
3957 017540 177777 177777  
3958 017544 177777 177777  
3959 017550 047447 047450  
3960 017554 000000  
3961  
3962  
3963  
3964  
3965  
3966  
3967 017556 000004  
3968 017560 012705 017572  
3969 017564 004737 023240  
3970  
3971 017570 000407  
3972  
3973 017572  
3974 017572 000200 000000  
3975 017576 000200 000000  
3976 017602 047517 047500  
3977 017606 000000  
3978  
3979  
3980  
3981  
3982  
3983  
3984 017610 000004  
3985 017612 012705 017624  
3986 017616 004737 023240  
3987  
3988 017622 000407  
3989  
3990 017624  
3991 017624 077777 177777  
3992 017630 077777 177777  
3993 017634 047557 047540  
3994 017640 000000  
3995  
3996  
3997  
3998  
3999  
4000  
4001 017642 000004  
4002 017644 012705 017656  
4003 017650 004737 023240  
4004  
4005 017654 000407  
4006

```

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        ;;

```

```

4007 017656
4008 017656 100200 000000
4009 017662 100200 000000
4010 017666 047407 047410
4011 017672 000000
4012
4013
4014
4015
4016
4017
4018 017674 000004
4019 017676 012705 017710
4020 017702 004737 023240
4021
4022 017706 000407
4023
4024 017710
4025 017710 000177 177777
4026 017714 000177 177777
4027 017720 047453 047444
4028 017724 000000
4029
4030
4031
4032
4033
4034
4035 017726 000004
4036 017730 012705 017742
4037 017734 004737 023240
4038
4039 017740 000407
4040
4041 017742
4042 017742 100000 000000
4043 017746 100000 000000
4044 017752 047503 147514
4045 017756 100014
4046
4047
4048
4049
4050
4051
4052 017760 000004
4053 017762 012705 017774
4054 017766 004737 023240
4055
4056 017772 000407
4057
4058 017774
4059 017774 100000 000001
4060 020000 100000 000001
4061 020004 047543 147554
4062 020010 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 M0,0 ; INITIAL MEM FLOAT NUMBER
        .WORD M0,0 ; 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 M0,1 ; INITIAL MEM FLOAT NUMBER
        .WORD M0,1 ; EXPECTED FLOAT RESULT
        .WORD 047543,147554 ; FPS: BEFORE, AFTER
        .WORD 100014 ; FEC AFTER ( 0 = N/A )

```



```

4063
4064
4065
4066
4067
4068
4069 020012 000004
4070 020014 012705 020026
4071 020020 004737 023240
4072
4073 020024 000407
4074
4075 020026
4076 020026 100177 177777
4077 020032 100177 177777
4078 020036 043403 043414
4079 020042 000000
4080
4081
4082

```

```

*****
*TEST 263 TEST OF TSTF INSTR, DATA SET TSTF-13
* ROUNDDING 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 )

```

```

4083
4084
4085
4086
4087 020044 000004
4088 020046 012705 020060
4089 020052 004737 023414
4090
4091 020056 000413
4092
4093 020060
4094 020060 000000 000000 000000
4095 020066 000000
4096 020070 003000 000000 000000
4097 020076 000000
4098 020100 047653 047644
4099 020104 000000
4100
4101
4102
4103
4104
4105
4106 020106 000004
4107 020110 012705 020122
4108 020114 004737 023414
4109
4110 020120 000413
4111
4112 020122
4113 020122 052525 052525 052525
4114 020130 052525
4115 020132 052525 052525 052525
4116 020140 052525
4117 020142 047717 047700
4118 020146 000000
4119
4120
4121
4122
4123
4124
4125 020150 000004
4126 020152 012705 020164
4127 020156 004737 023414
4128
4129 020162 000413
4130
4131 020164
4132 020164 125252 125252 125252
4133 020172 125252
4134 020174 125252 125252 125252
4135 020202 125252
4136 020204 047747 047750
4137 020210 000000
4138

```

```

*****
*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 )

```

```

4139
4140
4141
4142
4143
4144 020212 000004
4145 020214 012705 020226
4146 020220 004737 023414
4147
4148 020224 000413
4149
4150 020226
4151 020226 177777 177777 177777
4152 020234 177777
4153 020236 177777 177777 177777
4154 020244 177777
4155 020246 047607 047610
4156 020252 000000
4157
4158
4159
4160
4161
4162
4163 020254 000004
4164 020256 012705 020270
4165 020262 004737 023414
4166
4167 020266 000413
4168
4169 020270
4170 020270 000200 000000 000000
4171 020276 000000
4172 020300 000200 000000 000000
4173 020306 000000
4174 020310 047657 047640
4175 020314 000000
4176
4177
4178
4179
4180
4181
4182 020316 000004
4183 020320 012705 020332
4184 020324 004737 023414
4185
4186 020330 000413
4187
4188 020332
4189 020332 077777 177777 177777
4190 020340 177777
4191 020342 077777 177777 177777
4192 020350 177777
4193 020352 047717 047700
4194 020356 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,@TSTD1 ; GO TEST
BR TST270 ;;

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,@TSTD1 ; GO TEST
BR TST271 ;;

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,@TSTD1 ; GO TEST
BR TST272 ;;

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 )

```

4195  
4196  
4197  
4198  
4199  
4200  
4201 020360 000004  
4202 020362 012705 020374  
4203 020366 004737 023414  
4204  
4205 020372 000413  
4206  
4207 020374  
4208 020374 100200 000000 000000  
4209 020402 000000  
4210 020404 100200 000000 000000  
4211 020412 000000  
4212 020414 047747 047750  
4213 020420 000000  
4214  
4215  
4216  
4217  
4218  
4219  
4220 020422 000004  
4221 020424 012705 020436  
4222 020430 004737 023414  
4223  
4224 020434 000413  
4225  
4226 020436  
4227 020436 000177 177777 177777  
4228 020444 177777  
4229 020446 000177 177777 177777  
4230 020454 177777  
4231 020456 047613 047604  
4232 020462 000000  
4233  
4234  
4235  
4236  
4237  
4238  
4239 020464 000004  
4240 020466 012705 020500  
4241 020472 004737 023414  
4242  
4243 020476 000413  
4244  
4245 020500  
4246 020500 100000 000000 000000  
4247 020506 000000  
4248 020510 100000 000000 000000  
4249 020516 000000  
4250 020520 047643 147654

```

*****
*TEST 272      TEST OF TSTD INSTR, DATA SET TSTD-7
*              TRUNCATE MODE, ALL INTERRUPT ENABLES ON
*****

```

```

TSTD7: 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
*****

```

```

TSTD10: 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
*****

```

```

TSTD11: 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

```

4251 020524 100014 .WORD 100014 ; FEC AFTER ( 0 = N/A )

4252  
4253

4254 :\*\*\*\*\*  
4255 :\*TEST 275 TEST OF TSTD INSTR, DATA SET TSTD-12  
4256 :\* ROUNDING MODE, ALL INTERRUPT ENABLES ON  
4257 :\*\*\*\*\*

4258 020526 000004  
4259 020530 012705 020542  
4260 020534 004737 023414  
4261  
4262 020540 000413  
4263  
4264 020542  
4265 020542 100000 000000 000000  
4266 020550 000001  
4267 020552 100000 000000 000000  
4268 020560 000001  
4269 020562 047703 147714  
4270 020566 100014  
4271  
4272  
4273  
4274  
4275  
4276

TSTD12: ; TEST DATA SET TSTD-12:  
MOV #TSTD12,R5 ; PTR TO TEST DATA SET  
JSR PC,@TSTD12 ; GO TEST  
BR TSTD12 ; ;  
.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 )

4264

4265 020542 100000 000000 000000  
4266 020550 000001  
4267 020552 100000 000000 000000  
4268 020560 000001  
4269 020562 047703 147714  
4270 020566 100014

4271  
4272

4273 :\*\*\*\*\*  
4274 :\*TEST 276 TEST OF TSTD INSTR, DATA SET TSTD-13  
4275 :\* TRUNCATE MODE, -0 INTERRUPT ENABLE OFF, ALL OTHERS ON  
4276 :\*\*\*\*\*

4277 020570 000004  
4278 020572 012705 020604  
4279 020576 004737 023414  
4280  
4281 020602 000413  
4282  
4283 020604  
4284 020604 100177 177777 177777  
4285 020612 177777  
4286 020614 100177 177777 177777  
4287 020622 177777  
4288 020624 043643 043654  
4289 020630 000000  
4290  
4291  
4292

TSTD13: ; TEST DATA SET TSTD-13:  
MOV #TSTD13,R5 ; PTR TO TEST DATA SET  
JSR PC,@TSTD13 ; GO TEST  
BR TSTD13 ; ;  
.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 )

4283

4284 020604 100177 177777 177777  
4285 020612 177777  
4286 020614 100177 177777 177777  
4287 020622 177777  
4288 020624 043643 043654  
4289 020630 000000

4290  
4291  
4292

```

4293
4294
4295
4296 020632
4297 020632 000004
4298 020634 005037 001310
4299 020640 005037 001104
4300 020644 005037 001102
4301
4302
4303
4304
4305
4306
4307
4308
4309
4310
4311
4312 020650 076600 000022
4313 020654 032700 000020
4314 020660 001423
4315
4316 020662 032777 000002 160254
4317 020670 001017
4318
4319 020672 012701 010000
4320 020676 076600 000144
4321 020702 030100
4322 020704 001402
4323 020706 040100
4324 020710 000401
4325 020712 050100
4326 020714 076600 000344
4327
4328 020720 030100
4329 020722 001002
4330 020724 000137 003472
4331
4332
4333
4334
4335
4336
4337
4338
4339
4340
4341
4342
4343 020730
4344 020730 005037 001104
4345 020734 005037 001102
4346 020740 005037 001310
4347 020744 005237 001332
4348 020750 042737 100000 001332

```

```

*****
.SBTTL SUB PASS END CONTROL

```

```

TST277: .FORCE LAST TEST NUMBER
SCOPE ;CHECK FOR TEST ITERATIONS HERE
CLR $TIMES ;DONT ITERATE THIS "TEST"
CLR $ERFLG ;NO ERRORS HERE
CLR $TSTNM ;ZAP TEST ## WHEN DONE WITH A PASS

```

```

;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 $WHAMI ;GET WHAMI INTO RO
BIT $BIT04,RO ;1=HFP PRESENT, 0=NONE
BEQ $EOP ;EXIT IF NONE

BIT $SW01,$SWR ;1=HFP OR WFP TEST ONLY
BNE $EOP ;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 $2 ;
1$: BIS R1,RO ;SET BIT 12
2$: MED $WFLAG ;REWRITE FLAGS

BIT R1,RO ;HFP OR WFP NEXT ?
BNE $EOP ;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

```

```

$EOP:
CLR $ERFLG ;ZERO ERROR COUNT
CLR $TSTNM ;ZERO TEST NUMBER
CLR $TIMES ;ZERO NUMBER OF ITERATIONS
INC $PASS ;INCREMENT PASS COUNT,
BIC $100000,$PASS ; BUT NEVER LET IN GO NEGATIVE

```



```

4371 .SBTTL SUBR TO PERFORM TEST OF LDD/STD
4372
4373 LDADT:
4374 MOV #13,R0 ; LOAD STMP0-12
4375 MOV R5,R1 ; WITH TEST DATA SETS
4376 MOV #STMP0,R2 ; FOR DISPLAY LATER
4377 MOV (R1)+,(R2)+
4378 SOB R0,-2
4379 MOV #LDADL,$LPERR ; ERROR LOOPING ADDRESS
4380
4381 LDADL: LDFPS 20(R5) ; INITIAL FPS
4382
4383 JSR PC,(R4) ; GO PERFORM LDD/STD SEQUENCE
4384
4385 STFPS FPS ; STORE FPS AFTER
4386 STST FEC ; STORE FEC/FEA AFTER
4387
4388 CMP FPS,22(R5) ; CHECK FPS
4389 BEQ 65$ ; FPS IS OK
4390 ERROR 77 ; FPS BAD
4391 65$: TST 24(R5) ; DOES FEC/FEA APPLY?
4392 BPL 66$ ; NO - SKIP TEST
4393 MOV R4,EXPFEA ; GET EXPECTED FEA
4394 ADD #4,EXPFEA ; AND ADJUST
4395 CMPB FEC,24(R5) ; COMPARE FEC-S
4396 BNE 64$ ; NOT EQUAL
4397 CMP FEA,EXPFEA ; COMPARE FEA-S
4398 BEQ 66$ ; FEC, FEA OK
4399 64$: ERROR 102 ; FEC OR FEA ARE BAD
4400 66$:
4401
4402 CMP $REG0,10(R5) ; 1ST WORD OF RESULT CHECK?
4403 BNE 67$ ; NO
4404 CMP $REG1,12(R5) ; 2ND WORD OF RESULT CHECK?
4405 BNE 67$ ; NO
4406 CMP $REG2,14(R5) ; 3RD WORD OF RESULT CHECK?
4407 BNE 67$ ; NO
4408 CMP $REG3,16(R5) ; 4TH WORD OF RESULT CHECK?
4409 BEQ 68$ ; ALL WORDS OK
4410 67$: ERROR 104 ; NUMBERS NOT EQUAL
4411 68$:
4412
4413 RTS PC ; RETURN TO TEST CALLER
4414
4415 ; * * * LDD/STD SEQUENCE SUBR * * * * *
4416
4417 LDARDO: LDD PREVAC,AC0 ; PREV AC0 CONTENTS
4418 LDD (R5),AC0 ; LOAD AC0
4419 STD AC0,$REG0 ; STORE AC0
4420 RTS PC
4421 LDARD1: LDD PREVAC,AC1 ; PREV AC1 CONTENTS
4422 LDD (R5),AC1 ; LOAD AC1
4423 STD AC1,$REG0 ; STORE AC1
4424 RTS PC
4425 LDARD2: LDD PREVAC,AC2 ; PREV AC2 CONTENTS
4426 LDD (R5),AC2 ; LOAD AC2

```



4427	021252	174237	001170		STD	AC2, \$REGO	:	STORE AC2
4428	021256	000207			RTS	PC	:	
4429	021260	172737	002366	LDARD3:	LDD	PREVAC, AC3	:	PREV AC3 CONTENTS
4430	021264	172715			LDD	(R5), AC3	:	LOAD AC3
4431	021266	174337	001170		STD	AC3, \$REGO	:	STORE AC3
4432	021272	000207			RTS	PC	:	
4433	021274	172437	002366	LDARD4:	LDD	PREVAC, AC0	:	PREV AC0 CONTENTS
4434	021300	172415			LDD	(R5), AC0	:	INTERMEDIATE
4435	021302	174004			STD	AC0, AC4	:	LOAD AC4
4436	021304	172504			LDD	AC4, AC1	:	INTERMEDIATE
4437	021306	174137	001170		STD	AC1, \$REGO	:	STORE AC4
4438	021312	000207			RTS	PC	:	
4439	021314	172537	002366	LDARD5:	LDD	PREVAC, AC1	:	PREV AC1 CONTENTS
4440	021320	172515			LDD	(R5), AC1	:	INTERMEDIATE
4441	021322	174105			STD	AC1, AC5	:	LOAD AC5
4442	021324	172405			LDD	AC5, AC0	:	INTERMEDIATE
4443	021326	174037	001170		STD	AC0, \$REGO	:	STORE AC5
4444	021332	000207			RTS	PC	:	
4445							:	

SUBR TO PERFORM TEST OF LDD/LDF/STD

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

```

4446
4447
4448 021334
4449 021334 012700 000015
4450 021340 010501
4451 021342 012702 001230
4452 021346 012122
4453 021350 077002
4454 021352 012737 021360 001112
4455
4456 021360 170011
4457
4458 021362 004714
4459
4460 021364 170237 002330
4461 021370 170337 002332
4462
4463 021374 023765 002330 000026
4464 021402 001401
4465 021404 104100
4466 021406 005765 000030
4467 021412 100016
4468 021414 010437 002344
4469 021420 062737 000006 002344
4470 021426 123765 002332 000030
4471 021434 001004
4472 021436 023737 002334 002344
4473 021444 001401
4474 021446 104103
4475 021450
4476
4477 021450 023765 001170 000014
4478 021456 001014
4479 021460 023765 001172 000016
4480 021466 001010
4481 021470 023765 001174 000020
4482 021476 001004
4483 021500 023765 001176 000022
4484 021506 001401
4485 021510 104105
4486 021512
4487
4488 021512 000207
4489
4490
4491
4492 021514 172415
4493 021516 170165 000024
4494 021522 172465 000010
4495 021526 170011
4496 021530 174037 001170
4497 021534 000207
4498 021536 172515
4499 021540 170165 000024
4500 021544 172565 000010
4501 021550 170011

LDAFT:
MOV #15,R0 ; LOAD $TMPD-14
MOV R5,R1 ; WITH TEST DATA SETS
MOV #TMPD,R2 ; FOR DISPLAY LATER
MOV (R1)+,(R2)+ ;
SOB R0,-2 ;
MOV #LDAFL,$LPERR ; ERROR LOOPING ADDRESS

LDAFL:
SETD ; D MODE FOR INITIAL LOAD
JSR PC,(R4) ; GO PERFORM LDD/LDF/STD SEQUENCE
STFPS FPS ; STORE FPS AFTER
STST FEC ; STORE FEC/FEA AFTER
CMP FPS,26(R5) ; CHECK FPS
BEQ 65$ ; FPS IS OK
ERROR 100 ; FPS BAD
TST 30(R5) ; DOES FEC/FEA APPLY?
BPL 66$ ; NO - SKIP TEST
MOV R4,EXPFEA ; GET EXPECTED FEA
ADD #6,EXPFEA ; AND ADJUST
CMPB FEC,30(R5) ; COMPARE FEC-S
BNE 64$ ; NOT EQUAL
CMP FEA,EXPFEA ; COMPARE FEA-S
BEQ 66$ ; FEC, FEA OK
ERROR 103 ; FEC OR FEA ARE BAD
64$:
66$:

CMP $REG0,14(R5) ; 1ST WORD OF RESULT CHECK?
BNE 67$ ; NO
CMP $REG1,16(R5) ; 2ND WORD OF RESULT CHECK?
BNE 67$ ; NO
CMP $REG2,20(R5) ; 3RD WORD OF RESULT CHECK?
BNE 67$ ; NO
CMP $REG3,22(R5) ; 4TH WORD OF RESULT CHECK?
BEQ 68$ ; ALL WORDS OK
ERROR 105 ; NUMBERS NOT EQUAL
67$:
68$:

RTS PC ; RETURN TO TEST CALLER

; * * * LDD/LDF/STD SEQUENCE SUBR * * * * *
LDARF0:
LDD (R5),AC0 ; PREV AC0 CONTENTS
LDFPS 24(R5) ; INITIAL FPS
LDF 10(R5),AC0 ; LOAD NEW HALF OF AC0
SETD ; D MODE FOR STORE
STD AC0,$REG0 ; STORE ALL OF RESULT
RTS PC ;
LDARF1:
LDD (R5),AC1 ; PREV AC1 CONTENTS
LDFPS 24(R5) ; INITIAL FPS
LDF 10(R5),AC1 ; LOAD NEW HALF OF AC1
SETD ; D MODE FOR STORE

```

4502	021552	174137	001170		STD	AC1, \$REGO	:	STORE ALL OF RESULT
4503	021556	000207			RTS	PC	:	
4504	021560	172615		LDARF2:	LDD	(R5), AC2	:	PREV AC2 CONTENTS
4505	021562	170165	000024		LDFPS	24(R5)	:	INITIAL FPS
4506	021566	172665	000010		LDF	10(R5), AC2	:	LOAD NEW HALF OF AC2
4507	021572	170011			SETD		:	D MODE FOR STORE
4508	021574	174237	001170		STD	AC2, \$REGO	:	STORE ALL OF RESULT
4509	021600	000207			RTS	PC	:	
4510	021602	172715		LDARF3:	LDD	(R5), AC3	:	PREV AC3 CONTENTS
4511	021604	170165	000024		LDFPS	24(R5)	:	INITIAL FPS
4512	021610	172765	000010		LDF	10(R5), AC3	:	LOAD NEW HALF OF AC3
4513	021614	170011			SETD		:	D MODE FOR STORE
4514	021616	174337	001170		STD	AC3, \$REGO	:	STORE ALL OF RESULT
4515	021622	000207			RTS	PC	:	
4516	021624	172415		LDARF4:	LDD	(R5), AC0	:	INTERMEDIATE
4517	021626	174004			STD	AC0, AC4	:	PREV AC4 CONTENTS
4518	021630	170165	000024		LDFPS	24(R5)	:	INITIAL FPS
4519	021634	172465	000010		LDF	10(R5), AC0	:	INTERMEDIATE
4520	021640	174004			STF	AC0, AC4	:	LOAD NEW HALF OF AC4
4521	021642	170011			SETD		:	D MODE FOR STORE
4522	021644	172404			LDD	AC4, AC0	:	STORE ALL OF RESULT
4523	021646	174037	001170		STD	AC0, \$REGO	:	INTERMEDIATE
4524	021652	000207			RTS	PC	:	
4525	021654	172515		LDARF5:	LDD	(R5), AC1	:	INTERMEDIATE
4526	021656	174105			STD	AC1, AC5	:	PREV AC5 CONTENTS
4527	021660	170165	000024		LDFPS	24(R5)	:	INITIAL FPS
4528	021664	172565	000010		LDF	10(R5), AC1	:	INTERMEDIATE
4529	021670	174105			STF	AC1, AC5	:	LOAD NEW HALF OF AC5
4530	021672	170011			SETD		:	D MODE FOR STORE
4531	021674	172505			LDD	AC5, AC1	:	STORE ALL OF RESULT
4532	021676	174137	001170		STD	AC1, \$REGO	:	INTERMEDIATE
4533	021702	000207			RTS	PC	:	

```

4534 .SBTTL SUBR TO TEST THE ABSF INSTRUCTION
4535
4536 021704 ABSFT: MOV #7,R0 ; LOAD $TMPO-6
4537 021704 012700 000007 MOV R5,R1 ; WITH TEST DATA SETS
4538 021710 010501 MOV #TMPO,R2 ; FOR DISPLAY LATER
4539 021712 012702 001230 MOV (R1)+,(R2)+ ;
4540 021716 012122 MOV SOB RO,-2 ;
4541 021720 077002 MOV #ABSFL,$LPERR ; ERROR LOOPING ADDRESS
4542 021722 012737 021730 001112
4543
4544 021730 170001 ABSFL: SETF ; F MODE
4545 021732 011537 001170 MOV (R5),$REGO ; INITIAL FLOAT NUMBER
4546 021736 016537 000002 001172 MOV 2(R5),$REG1 ; INTO OPERATING ROOM
4547 021744 170165 000010 LDFPS 10(R5) ; INITIAL FPS
4548
4549 021750 170637 001170 ABSFI: ABSF $REGO ; ABS(($REGO))->$REGO
4550
4551 021754 170237 002330 STFPS FPS ; STORE FPS AFTER
4552 021760 170337 002332 STST FEC ; STORE FEC/FEA AFTER
4553
4554 021764 023765 002330 000012 CMP FPS,12(R5) ; CHECK FPS
4555 021772 001401 BEQ 65$ ; FPS IS OK
4556 021774 104076 ERROR 76 ; FPS BAD
4557 021776 005765 000014 65$: TST 14(R5) ; DOES FEC/FEA APPLY?
4558 022002 100014 BPL 66$ ; NO - SKIP TEST
4559 022004 012737 021750 002344 MOV #ABSFI,EXPFEA ; GET EXPECTED FEA
4560 022012 123765 002332 000014 CMPB FEC,14(R5) ; COMPARE FEC-S
4561 022020 001004 BNE 64$ ; NOT EQUAL
4562 022022 023737 002334 002344 CMP FEA,EXPFEA ; COMPARE FEA-S
4563 022030 001401 BEQ 66$ ; FEC, FEA OK
4564 022032 104101 64$: ERROR 101 ; FEC OR FEA ARE BAD
4565 022034 66$:
4566
4567 022034 023765 001170 000004 CMP $REGO,4(R5) ; 1ST WORD OF RESULT CHECK?
4568 022042 001004 BNE 67$ ; NO
4569 022044 023765 001172 000006 CMP $REG1,6(R5) ; 2ND WORD OF RESULT CHECK?
4570 022052 001401 BEQ 68$ ; ALL WORDS OK
4571 022054 104106 67$: ERROR 106 ; NUMBERS NOT EQUAL
4572 022056 68$:
4573
4574 022056 000207 RTS PC ; RETURN TO TEST CALLER
4575
4576 ;:*****
4577 .SBTTL SUBR TO TEST THE ABSD INSTRUCTION
4578
4579 022060 ABSDT: MOV #13,R0 ; LOAD $TMPO-12
4580 022060 012700 000013 MOV R5,R1 ; WITH TEST DATA SETS
4581 022064 010501 MOV #TMPO,R2 ; FOR DISPLAY LATER
4582 022066 012702 001230 MOV (R1)+,(R2)+ ;
4583 022072 012122 MOV SOB RO,-2 ;
4584 022074 077002 MOV #ABSDL,$LPERR ; ERROR LOOPING ADDRESS
4585 022076 012737 022104 001112
4586
4587 022104 170011 ABSDL: SETD ; D MODE
4588 022106 011537 001170 MOV (R5),$REGO ; INITIAL FLOAT NUMBER
4589 022112 016537 000002 001172 MOV 2(R5),$REG1 ; INTO OPERATING ROOM
  
```

```

4590 022120 016537 000004 001174      MOV      4(R5), $REG2      ;
4591 022126 016537 000006 001176      MOV      6(R5), $REG3      ;
4592 022134 170165 000020                LDFPS   20(R5)           ; INITIAL FPS
4593
4594 022140 170637 001170      ABSDI:  ABSD      $REG0      ; ABS(($REG0))->$REG0
4595
4596 022144 170237 002330      STFPS   FPS           ; STORE FPS AFTER
4597 022150 170337 002332      STST    FEC           ; STORE FEC/FEA AFTER
4598
4599 022154 023765 002330 000022      CMP     FPS, 22(R5)      ; CHECK FPS
4600 022162 001401                BEQ     65$             ; FPS IS OK
4601 022164 104077                ERROR   77             ; FPS BAD
4602 022166 005765 000024      65$:   TST    24(R5)      ; DOES FEC/FEA APPLY?
4603 022172 100014                BPL     66$             ; NO - SKIP TEST
4604 022174 012737 022140 002344      MOV     @ABSDI, EXPFEA    ; GET EXPECTED FEA
4605 022202 123765 002332 000024      CMPB   FEC, 24(R5)      ; COMPARE FEC-S
4606 022210 001004                BNE     64$             ; NOT EQUAL
4607 022212 023737 002334 002344      CMP     FEA, EXPFEA      ; COMPARE FEA-S
4608 022220 001401                BEQ     66$             ; FEC, FEA OK
4609 022222 104102                64$:   ERROR   102      ; FEC OR FEA ARE BAD
4610 022224
4611
4612 022224 023765 001170 000010      CMP     $REG0, 10(R5)    ; 1ST WORD OF RESULT CHECK?
4613 022232 001014                BNE     67$             ; NO
4614 022234 023765 001172 000012      CMP     $REG1, 12(R5)    ; 2ND WORD OF RESULT CHECK?
4615 022242 001010                BNE     67$             ; NO
4616 022244 023765 001174 000014      CMP     $REG2, 14(R5)    ; 3RD WORD OF RESULT CHECK?
4617 022252 001004                BNE     67$             ; NO
4618 022254 023765 001176 000016      CMP     $REG3, 16(R5)    ; 4TH WORD OF RESULT CHECK?
4619 022262 001401                BEQ     68$             ; ALL WORDS OK
4620 022264 104107                67$:   ERROR   107      ; NUMBERS NOT EQUAL
4621 022266
4622
4623 022266 000207                RTS     PC              ; RETURN TO TEST CALLER
4624

```

SUBR TO TEST THE NEGF INSTRUCTION

.SBTTL SUBR TO TEST THE NEGF INSTRUCTION

```

4625
4626
4627 022270
4628 022270 012700 000007
4629 022274 010501
4630 022276 012702 001230
4631 022302 012122
4632 022304 077002
4633 022306 012737 022314 001112
4634
4635 022314 170001
4636 022316 011537 001170
4637 022322 016537 000002 001172
4638 022330 170165 000010
4639
4640 022334 170737 001170
4641
4642 022340 170237 002330
4643 022344 170337 002332
4644
4645 022350 023765 002330 000012
4646 022356 001401
4647 022360 104076
4648 022362 005765 000014
4649 022366 100014
4650 022370 012737 022334 002344
4651 022376 123765 002332 000014
4652 022404 001004
4653 022406 023737 002334 002344
4654 022414 001401
4655 022416 104101
4656 022420
4657
4658 022420 023765 001170 000004
4659 022426 001004
4660 022430 023765 001172 000006
4661 022436 001401
4662 022440 104110
4663 022442
4664
4665 022442 000207
4666
4667
4668
4669
4670 022444
4671 022444 012700 000013
4672 022450 010501
4673 022452 012702 001230
4674 022456 012122
4675 022460 077002
4676 022462 012737 022470 001112
4677
4678 022470 170011
4679 022472 011537 001170
4680 022476 016537 000002 001172

```

NEGFT: MOV #7,R0 ; LOAD STMPO-6  
MOV R5,R1 ; WITH TEST DATA SETS  
MOV #STMPO,R2 ; FOR DISPLAY LATER  
MOV (R1)+,(R2)+  
SOB RO,-2  
MOV #NEGFL,SLPERR ; ERROR LOOPING ADDRESS  
NEGFL: SETF ; F MODE  
MOV (R5),SREG0 ; INITIAL FLOAT NUMBER  
MOV 2(R5),SREG1 ; INTO OPERATING ROOM  
LDFPS 10(R5) ; INITIAL FPS  
NEGFI: NEGF SREG0 ; -(SREG0)->SREG0  
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  
65\$: TST 14(R5) ; DOES FEC/FEA APPLY?  
BPL 66\$ ; NO - SKIP TEST  
MOV #NEGFI,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  
64\$: ERROR 101 ; FEC OR FEA ARE BAD  
66\$:  
CMP SREG0,4(R5) ; 1ST WORD OF RESULT CHECK?  
BNE 67\$ ; NO  
CMP SREG1,6(R5) ; 2ND WORD OF RESULT CHECK?  
BEQ 68\$ ; ALL WORDS OK  
67\$: ERROR 110 ; NUMBERS NOT EQUAL  
68\$:  
RTS PC ; RETURN TO TEST CALLER  
;:\*\*\*\*\*  
.SBTTL SUBR TO TEST THE NEG0 INSTRUCTION  
NEGDT: MOV #13,R0 ; LOAD STMPO-12  
MOV R5,R1 ; WITH TEST DATA SETS  
MOV #STMPO,R2 ; FOR DISPLAY LATER  
MOV (R1)+,(R2)+  
SOB RO,-2  
MOV #NEGDL,SLPERR ; ERROR LOOPING ADDRESS  
NEGDL: SETD ; D MODE  
MOV (R5),SREG0 ; INITIAL FLOAT NUMBER  
MOV 2(R5),SREG1 ; INTO OPERATING ROOM



```

4716 .SBTTL SUBR TO TEST THE CLRF INSTRUCTION
4717
4718 022654          012700 000007          CLRFT:  MOV    #7,R0          ; LOAD STMP0-6
4719 022654          010501 000007          MOV    R5,R1          ; WITH TEST DATA SETS
4720 022660          012702 001230          MOV    #STMP0,R2      ; FOR DISPLAY LATER
4721 022662          012122 000000          MOV    (R1)+,(R2)+   ;
4722 022666          077002 000000          SOB    R0,-2          ;
4723 022670          012737 022700 001112          MOV    #CLRFL,$LPERR ; ERROR LOOPING ADDRESS
4724 022672
4725
4726 022700          011537 001170          CLRFL:  SETF   (R5),$REG0 ; F MODE
4727 022702          016537 000002 001172          MOV    2(R5),$REG1   ; INITIAL FLOAT NUMBER
4728 022706          170165 000010          LDFPS 10(R5)         ; INTO OPERATING ROOM
4729 022714
4730
4731 022720          170437 001170          CLRFI:  CLRF   $REG0   ; 0->$REG0
4732
4733 022724          170237 002330          STFPS  FPS         ; STORE FPS AFTER
4734 022730          170337 002332          STST   FEC         ; STORE FEC/FEA AFTER
4735
4736 022734          023765 002330 000012          CMP    FPS,12(R5)    ; CHECK FPS
4737 022742          001401 000000          BEQ    65$          ; FPS IS OK
4738 022744          104076 000000          ERROR  76          ; FPS BAD
4739 022746          005765 000014          65$:   TST    14(R5)    ; DOES FEC/FEA APPLY?
4740 022752          100014 000000          BPL    66$          ; NO - SKIP TEST
4741 022754          012737 022720 002344          MOV    #CLRFI,EXPFEA ; GET EXPECTED FEA
4742 022762          123765 002332 000014          CMPB   FEC,14(R5)    ; COMPARE FEC-S
4743 022770          001004 000000          BNE    64$          ; NOT EQUAL
4744 022772          023737 002334 002344          CMP    FEA,EXPFEA   ; COMPARE FEA-S
4745 023000          001401 000000          BEQ    66$          ; FEC, FEA OK
4746 023002          104101 000000          64$:   ERROR  101     ; FEC OR FEA ARE BAD
4747 023004
4748
4749 023004          023765 001170 000004          CMP    $REG0,4(R5)   ; 1ST WORD OF RESULT CHECK?
4750 023012          001004 000000          BNE    67$          ; NO
4751 023014          023765 001172 000006          CMP    $REG1,6(R5)  ; 2ND WORD OF RESULT CHECK?
4752 023022          001401 000000          BEQ    68$          ; ALL WORDS OK
4753 023024          104112 000000          67$:   ERROR  112     ; NUMBERS NOT EQUAL
4754 023026
4755
4756 023026          000207          RTS    PC            ; RETURN TO TEST CALLER
4757
4758 ;:*****
4759 .SBTTL SUBR TO TEST THE CLRD INSTRUCTION
4760
4761 023030          012700 000013          CLRDT:  MOV    #13,R0     ; LOAD STMP0-12
4762 023030          010501 000013          MOV    R5,R1         ; WITH TEST DATA SETS
4763 023034          012702 001230          MOV    #STMP0,R2     ; FOR DISPLAY LATER
4764 023036          012122 000000          MOV    (R1)+,(R2)+   ;
4765 023042          077002 000000          SOB    R0,-2         ;
4766 023044          012737 023054 001112          MOV    #CLRD,$LPERR ; ERROR LOOPING ADDRESS
4767 023046
4768
4769 023054          011537 001170          CLRDL:  SETD   (R5),$REG0 ; D MODE
4770 023056          016537 000002 001172          MOV    2(R5),$REG1   ; INITIAL FLOAT NUMBER
4771 023062

```





```

4808 .SBTTL SUBR TO TEST THE TSTF INSTRUCTION
4809
4810 023240 TSTFT:
4811 023240 012700 000007 MOV #7,R0 ; LOAD $TMP0-6
4812 023244 010501 MOV R5,R1 ; WITH TEST DATA SETS
4813 023246 012702 001230 MOV #TMP0,R2 ; FOR DISPLAY LATER
4814 023252 012122 MOV (R1)+,(R2)+ ;
4815 023254 077002 SOB RO,-2 ;
4816 023256 012737 023264 001112 MOV #TSTFL,$LPERR ; ERROR LOOPING ADDRESS
4817
4818 023264 170001 TSTFL: SETF ; F MODE
4819 023266 011537 001170 MOV (R5),$REG0 ; INITIAL FLOAT NUMBER
4820 023272 016537 000002 001172 MOV 2(R5),$REG1 ; INTO OPERATING ROOM
4821 023300 170165 000010 LDFPS 10(R5) ; INITIAL FPS
4822
4823 023304 170537 001170 TSTFI: TSTF $REG0 ; ($REG0)->$REG0
4824
4825 023310 170237 002330 STFPS FPS ; STORE FPS AFTER
4826 023314 170337 002332 STST FEC ; STORE FEC/FEA AFTER
4827
4828 023320 023765 002330 000012 CMP FPS,12(R5) ; CHECK FPS
4829 023326 001401 BEQ 65$ ; FPS IS OK
4830 023330 104076 ERROR 76 ; FPS BAD
4831 023332 005765 000014 65$: TST 14(R5) ; DOES FEC/FEA APPLY?
4832 023336 100014 BPL 66$ ; NO - SKIP TEST
4833 023340 012737 023304 002344 MOV #TSTFI,EXPFEA ; GET EXPECTED FEA
4834 023346 123765 002332 000014 CMPB FEC,14(R5) ; COMPARE FEC-S
4835 023354 001004 BNE 64$ ; NOT EQUAL
4836 023356 023737 002334 002344 CMP FEA,EXPFEA ; COMPARE FEA-S
4837 023364 001401 BEQ 66$ ; FEC, FEA OK
4838 023366 104101 64$: ERROR 101 ; FEC OR FEA ARE BAD
4839 023370 66$:
4840
4841 023370 023765 001170 000004 CMP $REG0,4(R5) ; 1ST WORD OF RESULT CHECK?
4842 023376 001004 BNE 67$ ; NO
4843 023400 023765 001172 000006 CMP $REG1,6(R5) ; 2ND WORD OF RESULT CHECK?
4844 023406 001401 BEQ 68$ ; ALL WORDS OK
4845 023410 104114 67$: ERROR 114 ; NUMBERS NOT EQUAL
4846 023412 68$:
4847
4848 023412 000207 RTS PC ; RETURN TO TEST CALLER
4849
4850 ;:*****
4851 .SBTTL SUBR TO TEST THE TSTD INSTRUCTION
4852
4853 023414 TSTD:
4854 023414 012700 000013 MOV #13,R0 ; LOAD $TMP0-12
4855 023420 010501 MOV R5,R1 ; WITH TEST DATA SETS
4856 023422 012702 001230 MOV #TMP0,R2 ; FOR DISPLAY LATER
4857 023426 012122 MOV (R1)+,(R2)+ ;
4858 023430 077002 SOB RO,-2 ;
4859 023432 012737 023440 001112 MOV #TSTD,$LPERR ; ERROR LOOPING ADDRESS
4860
4861 023440 170011 TSTD: SETD ; D MODE
4862 023442 011537 001170 MOV (R5),$REG0 ; INITIAL FLOAT NUMBER
4863 023446 016537 000002 001172 MOV 2(R5),$REG1 ; INTO OPERATING ROOM

```



```

4899
4900
4901 023624 010637 002342
4902 023630 012637 002336
4903 023634 012637 002340
4904 023640 170237 002330
4905 023644 170337 002332
4906 023650 104116
4907 023652 013746 002340
4908 023656 013746 002336
4909 023662 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

.SBTTL SCOPE HANDLER ROUTINE

```

*****
;THIS ROUTINE CONTROLS THE LOOPING OF SUBTESTS. IT WILL INCREMENT
;AND LOAD THE TEST NUMBER($STNM) INTO THE DISPLAY REG.(DISPLAY<15:0>)
;THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
;SW14=1      LOOP ON TEST
;SW11=1      INHIBIT ITERATIONS
;SM09=1      LOOP ON ERROR
;SM08=1      LOOP ON TEST IN "$LPTST"
;CALL
;*          SCOPE          ;;SCOPE=IOT

```

```

4910
4911
4912
4913
4914
4915
4916
4917
4918
4919
4920
4921
4922
4923 023664
4924 023664
4925 023664 032777 040000 155252
4926 023672 001114
4927
4928 023674 000416
4929
4930 023676 013746 000004
4931 023702 012737 023722 000004
4932 023710 005737 177060
4933 023714 012637 000004
4934 023720 000463
4935 023722 022626
4936 023724 012637 000004
4937 023730 000423
4938 023732
4939 023732 032777 000400 155204
4940 023740 001404
4941 023742 023737 001150 001102
4942 023750 001465
4943 023752 005737 001104
4944 023756 001421
4945 023760 023737 001120 001104
4946 023766 101015
4947 023770 032777 001000 155146
4948 023776 001404
4949 024000 013737 001112 001110
4950 024006 000446
4951 024010 005037 001104
4952 024014 005037 001310
4953 024020 000415
4954 024022 032777 004000 155114
4955 024030 001011
4956 024032 005737 001332
4957 024036 001406
4958 024040 005237 001106
4959 024044 023737 001310 001106
4960 024052 002024
4961 024054 012737 000001 001106
4962 024062 013737 024140 001310
4963 024070 005237 001102
4964 024074 013737 001102 001330
4965 024102 011637 001110

```

```

$SCOPE:
64$:
1$: BIT #BIT14,$SWR ;;LOOP ON PRESENT TEST?
   BNE $OVER ;;YES IF SW14=1
;####START OF CODE FOR THE XOR TESTER####
$XTSTR: BR 6$ ;;IF RUNNING ON THE "XOR" TESTER CHANGE
   ;;THIS INSTRUCTION TO A "NOP" (NOP=240)
   MOV $#ERRVEC,-(SP) ;;SAVE THE CONTENTS OF THE ERROR VECTOR
   MOV #5,$#ERRVEC ;;SET FOR TIMEOUT
   TST $#177060 ;;TIME OUT ON XOR?
   MOV (SP)+,$#ERRVEC ;;RESTORE THE ERROR VECTOR
   BR $SVLAD ;;GO TO THE NEXT TEST
5$: CMP (SP)+,(SP)+ ;;CLEAR THE STACK AFTER A TIME OUT
   MOV (SP)+,$#ERRVEC ;;RESTORE THE ERROR VECTOR
   BR 7$ ;;LOOP ON THE PRESENT TEST
6$;####END OF CODE FOR THE XOR TESTER####
   BIT #BIT08,$SWR ;;LOOP ON SPEC. TEST?
   BEQ 2$ ;;BR IF NO
   CMP $LPTST,$STNM ;;ON THE RIGHT TEST?
   BEQ $OVER ;;BR IF YES
2$: TST $ERFLG ;;HAS AN ERROR OCCURRED?
   BEQ 3$ ;;BR IF NO
   CMP $ERMAX,$ERFLG ;;MAX. ERRORS FOR THIS TEST OCCURRED?
   BHI 3$ ;;BR IF NO
   BIT #BIT09,$SWR ;;LOOP ON ERROR?
   BEQ 4$ ;;BR IF NO
7$: MOV $LPERR,$LPADR ;;SET LOOP ADDRESS TO LAST SCOPE
   BR $OVER
4$: CLR $ERFLG ;;ZERO THE ERROR FLAG
   CLR $TIMES ;;CLEAR THE NUMBER OF ITERATIONS TO MAKE
   BR 1$ ;;ESCAPE TO THE NEXT TEST
3$: BIT #BIT11,$SWR ;;INHIBIT ITERATIONS?
   BNE 1$ ;;BR IF YES
   TST $PASS ;;IF FIRST PASS OF PROGRAM
   BEQ 1$ ;;INHIBIT ITERATIONS
   INC $ICNT ;;INCREMENT ITERATION COUNT
   CMP $TIMES,$ICNT ;;CHECK THE NUMBER OF ITERATIONS MADE
   BGE $OVER ;;BR IF MORE ITERATION REQUIRED
1$: MOV #1,$ICNT ;;REINITIALIZE THE ITERATION COUNTER
   MOV $MXCNT,$TIMES ;;SET NUMBER OF ITERATIONS TO DO
$SVLAD: INC $STNM ;;COUNT TEST NUMBERS
   MOV $STNM,$TESTN ;;SET TEST NUMBER IN APT MAILBOX
   MOV (SP),$LPADR ;;SAVE SCOPE LOOP ADDRESS

```

```

4966 024106 011637 001112          MOV      (SP) $LPERR      ;; SAVE ERROR LOOP ADDRESS
4967 024112 005037 001312          CLR      $ESCAPE        ;; CLEAR THE ESCAPE FROM ERROR ADDRESS
4968 024116 012737 000001 001120    MOV      #1, $ERMAX      ;; ONLY ALLOW ONE(1) ERROR ON NEXT TEST
4969 024124 013777 001102 155014    $OVER:  MOV      $STNM, $DISPLAY ;; DISPLAY TEST NUMBER
4970 024132 013716 001110          MOV      $LPADR, (SP)   ;; FUDGE RETURN ADDRESS
4971 024136 000002          RTI                    ;; FIXES PS
4972 024140 003720          $MXCNT: 2000.         ;; MAX. NUMBER OF ITERATIONS

```

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  
5010  
5011  
5012  
5013  
5014  
5015  
5016  
5017  
5018  
5019  
5020  
5021  
5022  
5023  
5024  
5025  
5026  
5027  
5028

024142  
024142 010037 002346  
024146 010137 002350  
024152 010237 002352  
024156 010337 002354  
024162 010437 002356  
024166 010537 002360  
024172 010637 002362  
024176 062737 000004 002362  
024204 011637 002364  
024210 005237 001104  
024214 001775  
024216 013777 001102 154722  
024224 032777 002000 154712  
024232 001402  
024234 104401 001314  
024240 005237 001114  
024244 011637 001122  
024250 162737 000002 001122  
024256 117737 154640 001116  
024264 032777 020000 154652  
024272 001004  
024274 004737 024404  
024300 104401 001321  
024304  
024304 122737 000001 001344  
024312 001007  
024314 113737 001116 024326  
024322 004737 025120  
024326 000  
024327 000  
024330 000777  
024332 005777 154606  
024336 100001  
024340 000000  
024342 032777 001000 154574  
024350 001402  
024352 013716 001112  
024356 005737 001312  
024362 001402  
024364 013716 001312  
024370

```
.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

2$:
TST @SWR ; HALT ON ERROR
BPL 3$ ; 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

5$:
```

K10

FPU BASIC INSTR TESTS MACY11 27(1006) 25-APR-77 09:12 PAGE 108  
DQFPAB.P11 19-APR-77 13:36 ERROR HANDLER ROUTINE

SEQ 0112

5029	024370	022737	021016	000042	CMP	#SENDAD, 0#42	:::ACT-11 AUTO-ACCEPT?
5030	024376	001001			BNE	6\$	:::BRANCH IF NO
5031	024400	000000			HALT		:::YES
5032	024402				6\$:		
5033	024402	000002			64\$:	RTI	;RETURN



```

5034
5035
5036
5037
5038
5039
5040
5041
5042
5043
5044
5045
5046
5047
5048 024404
5049 024404 104401
5050 024406 001321
5051 024410 010046
5052 024412 010146
5053 024414 005000
5054 024416 153700 001116
5055 024422 001004
5056
5057 024424 013746 001122
5058 024430 104402
5059 024432 000452
5060 024434 005300
5061 024436 006300
5062 024440 010001
5063 024442 006300
5064 024444 060100
5065 024446 062700 001354
5066 024452 012037 024462
5067 024456 001404
5068 024460 104401
5069 024462 000000
5070 024464 104401 001321
5071 024470 104401 024600
5072 024474 012037 024504
5073 024500 001402
5074 024502 104401
5075 024504 000000
5076 024506 104401 001321
5077 024512 017746 000054
5078 024516 104402
5079 024520 104401 024576
5080 024524 017746 000044
5081 024530 104402
5082 024532 104401 024576
5083 024536 011000
5084 024540 001407
5085 024542 013046
5086 024544 104402
5087 024546 005710
5088 024550 001403
5089 024552 104401 024576

```

```

;*****
.SBTTL ERROR MESSAGE TIMEOUT ROUTINE (MODIFIED SYSMAC)
;THIS ROUTINE USES THE "ITEM CONTROL BYTE" ($ITEMB) TO DETERMINE WHICH
;ERROR IS TO BE REPORTED. IT THEN OBTAINS, FROM THE "ERROR TABLE",
;($ERRTB) 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 $ERRPC AS THE FIRST TWO DATA ELEMENTS
;(WITH APPROPRIATE HEADERS).

```

```

$STYPERR:
HOTWRM: .WORD $CRLF ; TYPE "HOT" OR "WARM"
        MOV RO,-(SP) ; PTR TO MESSAGE
        MOV R1,-(SP) ; SAVE R0
        CLR RO ; SAVE R1
        BISB @#$ITEMB,RO ; PICKUP ITEM INDEX
        BNE 1$ ; IF ITEM NUMBER FROM ERROR 0,
        ; JUST TYPE PC OF ERROR
        MOV $ERRPC,-(SP) ; GET ERROR PC FOR TIMEOUT
        TYPOC ; TYPE OCTAL, ALL DIGITS
        BR 7$ ; EXIT
1$: DEC RO ; ADJUST ERROR # FOR TABLE INDEX
    ASL RO ; OF 6 BYTES/ENTRY
    MOV RO,R1
    ASL RO
    ADD R1,RO
    ADD @#$ERRTB,RO ; FORM TABLE PTR
    MOV (RO)+,2$ ; PICKUP "ERROR MESSAGE" PTR
    BEQ 3$ ; SKIP TIMEOUT IF NULL
    TYPE ; TYPE "ERROR MESSAGE"
    .WORD 0 ; "ERROR MESSAGE" PTR HERE
2$: .WORD 0 ; CR & LF
    TYPE , $CRLF ; "TEST # ERR PC" HEADER
    .WORD 11$ ; PICKUP "DATA HEADER" PTR
    MOV (RO)+,4$ ; SKIP TIMEOUT IF NULL
    BEQ 5$ ; TYPE "DATA HEADER"
    TYPE ; "DATA HEADER" PTR HERE
    .WORD 0 ; CR & LF
3$: TYPE $CRLF ; ($TESTN)
    MOV @8$,-(SP) ; OCTAL W/ LEADING ZEROS
    TYPOC ; <HT>
    TYPE 10$ ; ($ERRPC)
    MOV @9$,-(SP) ; OCTAL W/ LEADING ZEROS
    TYPOC ; <HT>
    MOV (RO),RO ; PICKUP "DATA TABLE" PTR
    BEQ 7$ ; EXIT IF NULL
4$: MOV @2(RO)+,-(SP) ; SAVE ... FOR TIMEOUT
    TYPOC ; TYPE OCTAL, ALL DIGITS
5$: TST (RO) ; ANOTHER NUMBER ?
    BEQ 7$ ; NO - EXIT
    TYPE ,10$ ; TAB BETWEEN ELEMENTS

```

```

5090 024556 000771
5091 024560 012601
5092 024562 012600
5093 024564 104401 001321
5094 024570 000207
5095 024572 001330
5096 024574 001122
5097 024576 000011
5098 024600 042524 052123 021440
5099 024606 042411 051122 050040
5100 024614 004503 000
5101 024620

7S: BR 6S
MOV (SP)+,R1
MOV (SP)+,R0
TYPE $CRLF
RTS PC
8S: .WORD $TESTN
9S: .WORD $ERRPC
10S: .ASCIZ <11>
11S: .ASCIZ "TEST # ERR PC

: LOOP ON DATA TABLE VECTOR
: RESTORE R1
: RESTORE R0
: CR & LF
: RETURN
: <HT>

.EVEN

```

5102  
5103  
5104  
5105  
5106  
5107  
5108  
5109  
5110  
5111  
5112  
5113  
5114  
5115  
5116  
5117  
5118  
5119  
5120  
5121  
5122  
5123  
5124  
5125  
5126  
5127  
5128  
5129  
5130  
5131  
5132  
5133  
5134  
5135  
5136  
5137  
5138  
5139  
5140  
5141  
5142  
5143  
5144  
5145  
5146  
5147  
5148  
5149  
5150  
5151  
5152  
5153  
5154  
5155  
5156  
5157

.SBTTL TYPE ROUTINE

\*\*\*\*\*  
\*ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.  
\*THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.  
\*NOTE1: \$NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.  
\*NOTE2: \$FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.  
\*NOTE3: \$FILLC CONTAINS THE CHARACTER TO FILL AFTER.  
\*

\*CALL:  
\*1) USING A TRAP INSTRUCTION  
\* TYPE ,MESADR ;;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING  
\*OR  
\* TYPE  
\* MESADR  
\*

024620	105737	001165	STYPE:	TSTB	\$TPFLG	:: IS THERE A TERMINAL?
024624	100002			BPL	1\$	:: BR IF YES
024626	000000			HALT		:: HALT HERE IF NO TERMINAL
024630	000430			BR	3\$	:: LEAVE
024632	010046		1\$:	MOV	RO, -(SP)	:: SAVE RO
024634	017600	000002		MOV	#2(SP), RO	:: GET ADDRESS OF ASCIZ STRING
024640	122737	000001 001344		CMPB	#APTENV, \$ENV	:: RUNNING IN APT MODE
024646	001011			BNE	62\$	:: NO GO CHECK FOR APT CONSOLE
024650	132737	000100 001345		BITB	#APTSPool, \$ENVM	:: SPOOL MESSAGE TO APT
024656	001405			BEQ	62\$	:: NO GO CHECK FOR CONSOLE
024660	010037	024670		MOV	RO, 61\$	:: SETUP MESSAGE ADDRESS FOR APT
024664	004737	025110		JSR	PC, \$ATY3	:: SPOOL MESSAGE TO APT
024670	000000		61\$:	.WORD	0	:: MESSAGE ADDRESS
024672	132737	000040 001345	62\$:	BITB	#APTCSUP, \$ENVM	:: APT CONSOLE SUPPRESSED
024700	001003			BNE	60\$	:: YES, SKIP TYPE OUT
024702	112046		2\$:	MOVB	(RO)+, -(SP)	:: PUSH CHARACTER TO BE TYPED ONTO STACK
024704	001005			BNE	4\$	:: BR IF IT ISN'T THE TERMINATOR
024706	005726			TST	(SP)+	:: IF TERMINATOR POP IT OFF THE STACK
024710	012600		60\$:	MOV	(SP)+, RO	:: RESTORE RO
024712	062716	000002	3\$:	ADD	#2, (SP)	:: ADJUST RETURN PC
024716	000002			RTI		:: RETURN
024720	122716	000011	4\$:	CMPB	#HT, (SP)	:: BRANCH IF <HT>
024724	001430			BEQ	8\$	
024726	122716	000200		CMPB	#CRLF, (SP)	:: BRANCH IF NOT <CRLF>
024732	001006			BNE	5\$	
024734	005726			TST	(SP)+	:: POP <CR><LF> EQUIV
024736	104401			TYPE		:: TYPE A CR AND LF
024740	001321			\$CRLF		
024742	105037	025076		CLRB	\$CHARCNT	:: CLEAR CHARACTER COUNT
024746	000755			BR	2\$	:: GET NEXT CHARACTER
024750	004737	025032	5\$:	JSR	PC, \$TYPEC	:: GO TYPE THIS CHARACTER
024754	123726	001164	6\$:	CMPB	\$FILLC, (SP)+	:: IS IT TIME FOR FILLER CHARS.?
024760	001350			BNE	2\$	:: IF NO GO GET NEXT CHAR.
024762	013746	001162		MOV	\$NULL, -(SP)	:: GET # OF FILLER CHARS. NEEDED
						:: AND THE NULL CHAR.
024766	105366	000001	7\$:	DECB	1(SP)	:: DOES A NULL NEED TO BE TYPED?
024772	002770			BLT	6\$	:: BR IF NO--GO POP THE NULL OFF OF STACK
024774	004737	025032		JSR	PC, \$TYPEC	:: GO TYPE A NULL
025000	105337	025076		DECB	\$CHARCNT	:: DO NOT COUNT AS A COUNT

```

5158 025004 000770          BR      7$          ;;LOOP
5159
5160          ;HORIZONTAL TAB PROCESSOR
5161
5162 025006 112716 000040      8$:   MOVB   #' (SP)          ;; REPLACE TAB WITH SPACE
5163 025012 004737 025032      9$:   JSR    PC,$TYPEC          ;; TYPE A SPACE
5164 025016 132737 000007 025076  BITB   #7,$SCHARCNT          ;; BRANCH IF NOT AT
5165 025024 001372          BNE    9$          ;; TAB STOP
5166 025026 005726          TST    (SP)+          ;; POP SPACE OFF STACK
5167 025030 000724          BR     2$          ;; GET NEXT CHARACTER
5168 025032 105777 154120      $TYPEC: TSTB  @STPS          ;; WAIT UNTIL PRINTER IS READY
5169 025036 100375          BPL   $TYPEC
5170 025040 116677 000002 154112  MOVB   2(SP),@STPB          ;; LOAD CHAR TO BE TYPED INTO DATA REG.
5171 025046 122766 000015 000002  CMPB   #CR,2(SP)          ;; IS CHARACTER A CARRIAGE RETURN?
5172 025054 001003          BNE    1$          ;; BRANCH IF NO
5173 025056 105037 025076          CLRB  $SCHARCNT          ;; YES--CLEAR CHARACTER COUNT
5174 025062 000406          BR    $TYPEX          ;; EXIT
5175 025064 122766 000012 000002  1$:   CMPB   #LF,2(SP)          ;; IS CHARACTER A LINE FEED?
5176 025072 001402          BEQ   $TYPEX          ;; BRANCH IF YES
5177 025074 105227          INCB  (PC)+          ;; COUNT THE CHARACTER
5178 025076 000000          $SCHARCNT: .WORD 0          ;; CHARACTER COUNT STORAGE
5179 025100 000207          $TYPEX: RTS   PC
5180

```

```

5181 .SBTTL APT COMMUNICATIONS ROUTINE
5182
5183 ..*****
5184 025102 112737 000001 025346 $ATY1: MOVB #1,$FFLG ;; TO REPORT FATAL ERROR
5185 025110 112737 000001 025344 $ATY3: MOVB #1,$MFLG ;; TO TYPE A MESSAGE
5186 025116 000403 BR $ATYC
5187 025120 112737 000001 025346 $ATY4: MOVB #1,$FFLG ;; TO ONLY REPORT FATAL ERROR
5188 025126 $ATYC:
5189 025126 010046 MOV RO,-(SP) ;; PUSH RO ON STACK
5190 025130 010146 MOV R1,-(SP) ;; PUSH R1 ON STACK
5191 025132 105737 025344 TSTB $MFLG ;; SHOULD TYPE A MESSAGE?
5192 025136 001450 BEQ 5$ ;; IF NOT: BR
5193 025140 122737 000001 001344 CMPB #APTENV,$ENV ;; OPERATING UNDER APT?
5194 025146 001031 BNE 3$ ;; IF NOT: BR
5195 025150 132737 000100 001345 BITB #APTSPool,$ENVM ;; SHOULD SPOOL MESSAGES?
5196 025156 001425 BEQ 3$ ;; IF NOT: BR
5197 025160 017600 000004 MOV #4(SP),RO ;; GET MESSAGE ADDR.
5198 025164 062766 000002 000004 ADD #2,4(SP) ;; BUMP RETURN ADDR.
5199 025172 005737 001324 1$: TST $MSGTYPE ;; SEE IF DONE W/ LAST XMISSION?
5200 025176 001375 BNE 1$ ;; IF NOT: WAIT
5201 025200 010037 001340 MOV RO,$MSGAD ;; PUT ADDR IN MAILBOX
5202 025204 105720 2$: TSTB (RO)+ ;; FIND END OF MESSAGE
5203 025206 001376 BNE 2$
5204 025210 163700 001340 SUB $MSGAD,RO ;; SUB START OF MESSAGE
5205 025214 006200 ASR RO ;; GET MESSAGE LNGTH IN WORDS
5206 025216 010037 001342 MOV RO,$MSGLG; ;; PUT LENGTH IN MAILBOX
5207 025222 012737 000004 001324 MOV #4,$MSGTYPE ;; TELL APT TO TAKE MSG.
5208 025230 000413 BR 5$
5209 025232 017637 000004 025256 3$: MOV #4(SP),4$ ;; PUT MSG ADDR IN JSR LINKAGE
5210 025240 062766 000002 000004 ADD #2,4(SP) ;; BUMP RETURN ADDRESS
5211 025246 013746 177776 MOV 177776,-(SP) ;; PUSH 177776 ON STACK
5212 025252 004737 024620 JSR PC,$TYPE ;; CALL TYPE MACRO
5213 025256 000000 4$: .WORD 0
5214 025260 5$:
5215 025260 105737 025346 10$: TSTB $FFLG ;; SHOULD REPORT FATAL ERROR?
5216 025264 001416 BEQ 12$ ;; IF NOT: BR
5217 025266 005737 001344 TST $ENV ;; RUNNING UNDER APT?
5218 025272 001413 BEQ 12$ ;; IF NOT: BR
5219 025274 005737 001324 11$: TST $MSGTYPE ;; FINISHED LAST MESSAGE?
5220 025300 001375 BNE 11$ ;; IF NOT: WAIT
5221 025302 017637 000004 001326 MOV #4(SP),$FATAL ;; GET ERROR #
5222 025310 062766 000002 000004 ADD #2,4(SP) ;; BUMP RETURN ADDR.
5223 025316 005237 001324 INC $MSGTYPE ;; TELL APT TO TAKE ERROR
5224 025322 105037 025346 12$: CLRB $FFLG ;; CLEAR FATAL FLAG
5225 025326 105037 025345 CLRB $LFLG ;; CLEAR LOG FLAG
5226 025332 105037 025344 CLRB $MFLG ;; CLEAR MESSAGE FLAG
5227 025336 012601 MOV (SP)+,R1 ;; POP STACK INTO R1
5228 025340 012600 MOV (SP)+,RO ;; POP STACK INTO RO
5229 025342 000207 RTS PC ;; RETURN
5230 025344 000 $MFLG: .BYTE 0 ;; MESSG. FLAG
5231 025345 000 $LFLG: .BYTE 0 ;; LOG FLAG
5232 025346 000 $FFLG: .BYTE 0 ;; FATAL FLAG
5233 025350 .EVEN
5234 000200 APTSIZE=200
5235 000001 APTENV=001
5236 000100 APTSPool=100

```

D11

FPU BASIC INSTR TESTS MACY11 27(1006) 25-APR-77 09:12 PAGE 114  
DQFPAB.P11 19-APR-77 13:36 APT COMMUNICATIONS ROUTINE

SEQ 0118

5237

000040

APTCSUP=040

.SBTTL BINARY TO OCTAL (ASCII) AND TYPE

5238  
5239  
5240  
5241  
5242  
5243  
5244  
5245  
5246  
5247  
5248  
5249  
5250  
5251  
5252  
5253  
5254  
5255  
5256  
5257  
5258  
5259  
5260  
5261  
5262  
5263 025350 017646 000000  
5264 025354 116637 000001 025573  
5265 025362 112637 025575  
5266 025366 062716 000002  
5267 025372 000406  
5268 025374 112737 000001 025573  
5269 025402 112737 000006 025575  
5270 025410 112737 000005 025572  
5271 025416 010346  
5272 025420 010446  
5273 025422 010546  
5274 025424 113704 025575  
5275 025430 005404  
5276 025432 062704 000006  
5277 025436 110437 025574  
5278 025442 113704 025573  
5279 025446 016605 000012  
5280 025452 005003  
5281 025454 006105 1\$:  
5282 025456 000404  
5283 025460 006105 2\$:  
5284 025462 006105  
5285 025464 006105  
5286 025466 010503  
5287 025470 006103 3\$:  
5288 025472 105337 025574  
5289 025476 100016  
5290 025500 042703 177770  
5291 025504 001002  
5292 025506 005704  
5293 025510 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
*      TYPON    ;;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
*STYON---ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST
*STYPOS OR STYOC
*CALL:
*      MOV      NUM,-(SP)      ;;NUMBER TO BE TYPED
*      TYPON    ;;CALL FOR TYPEOUT
*STYOC---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
*          MOVVB  1(SP),SOFILL    ;;LOAD ZERO FILL SWITCH
*          MOVVB  (SP)+,SOMODE+1  ;;NUMBER OF DIGITS TO TYPE
*          ADD    #2,(SP)        ;;ADJUST RETURN ADDRESS
*          BR     STYON
*STYOC:  MOVVB  #1,SOFILL        ;;SET THE ZERO FILL SWITCH
*          MOVVB  #6,SOMODE+1    ;;SET FOR SIX(6) DIGITS
*STYON:  MOVVB  #5,SOCNT        ;;SET THE ITERATION COUNT
*          MOV    R3,-(SP)      ;;SAVE R3
*          MOV    R4,-(SP)      ;;SAVE R4
*          MOV    R5,-(SP)      ;;SAVE R5
*          MOVVB  SOMODE+1,R4    ;;GET THE NUMBER OF DIGITS TO TYPE
*          NEG    R4            ;;SUBTRACT IT FOR MAX. ALLOWED
*          ADD    #6,R4        ;;SAVE IT FOR USE
*          MOVVB  R4,SOMODE     ;;GET THE ZERO FILL SWITCH
*          MOV    SOFILL,R4    ;;PICKUP THE INPUT NUMBER
*          MOV    12(SP),R5    ;;CLEAR THE OUTPUT WORD
*          CLR    R3          ;;ROTATE MSB INTO "C"
*          ROL   R5          ;;GO DO MSB
*          BR    3$          ;;FORM THIS DIGIT
*          BR    3$
*          ROL   R5
*          ROL   R5
*          ROL   R5
*          MOV   R5,R3
*          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

```

5294	025512	005204		4\$:	INC	R4	::	DON'T SUPPRESS ANYMORE D'S
5295	025514	052703	000060		BIS	#'0,R3	::	MAKE THIS DIGIT ASCII
5296	025520	052703	000040	5\$:	BIS	#',R3	::	MAKE ASCII IF NOT ALREADY
5297	025524	110337	025570		MOVB	R3,8\$	::	SAVE FOR TYPING
5298	025530	104401	025570		TYPE	8\$	::	GO TYPE THIS DIGIT
5299	025534	105337	025572	7\$:	DECB	\$OCNT	::	COUNT BY 1
5300	025540	003347			BGT	2\$	::	BR IF MORE TO DO
5301	025542	002402			BLT	6\$	::	BR IF DONE
5302	025544	005204			INC	R4	::	INSURE LAST DIGIT ISN'T A BLANK
5303	025546	000744			BR	2\$	::	GO DO THE LAST DIGIT
5304	025550	012605		6\$:	MOV	(SP)+,R5	::	RESTORE R5
5305	025552	012604			MOV	(SP)+,R4	::	RESTORE R4
5306	025554	012603			MOV	(SP)+,R3	::	RESTORE R3
5307	025556	016666	000002 000004		MOV	2(SP),4(SP)	::	SET THE STACK FOR RETURNING
5308	025564	012616			MOV	(SP)+,(SP)	::	
5309	025566	000002			RTI		::	RETURN
5310	025570	000		8\$:	.BYTE	0	::	STORAGE FOR ASCII DIGIT
5311	025571	000			.BYTE	0	::	TERMINATOR FOR TYPE ROUTINE
5312	025572	000		\$OCNT:	.BYTE	0	::	OCTAL DIGIT COUNTER
5313	025573	000		\$OFILL:	.BYTE	0	::	ZERO FILL SWITCH
5314	025574	000000		\$OMODE:	.WORD	0	::	NUMBER OF DIGITS TO TYPE



5315  
5316  
5317  
5318  
5319  
5320  
5321  
5322  
5323 025576 010046  
5324 025600 016600 000002  
5325 025604 005740  
5326 025606 111000  
5327 025610 006300  
5328 025612 016000 025632  
5329 025616 000200  
5330  
5331  
5332  
5333  
5334 025620 011646  
5335 025622 016666 000004 000002  
5336 025630 000002  
5337  
5338  
5339  
5340  
5341  
5342  
5343  
5344  
5345 025632 025620  
5346 025634 024620  
5347 025636 025374  
5348 025640 025350  
5349 025642 025410  
5350  
5351

.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)

5352  
5353  
5354  
5355  
5356  
5357  
5358  
5359  
5360  
5361  
5362  
5363  
5364  
5365  
5366  
5367  
5368  
5369  
5370  
5371  
5372  
5373  
5374  
5375  
5376  
5377  
5378  
5379  
5380  
5381  
5382  
5383  
5384  
5385  
5386  
5387  
5388  
5389  
5390  
5391  
5392  
5393  
5394  
5395  
5396  
5397  
5398

.SBTTL POWER DOWN AND UP ROUTINES

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

:POWER DOWN ROUTINE

```

$PWRDN: MOV    $SILLUP, @PWRVEC    ;; SET FOR FAST UP
        MOV    #340, @PWRVEC+2    ;; PRIO:7
        MOV    R0, -(SP)          ;; PUSH R0 ON STACK
        MOV    R1, -(SP)          ;; PUSH R1 ON STACK
        MOV    R2, -(SP)          ;; PUSH R2 ON STACK
        MOV    R3, -(SP)          ;; PUSH R3 ON STACK
        MOV    R4, -(SP)          ;; PUSH R4 ON STACK
        MOV    R5, -(SP)          ;; PUSH R5 ON STACK
        MOV    @SWR, -(SP)        ;; PUSH @SWR ON STACK
        MOV    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
        IS:   INC    $SAVR6       ;; WAIT FOR THE INC
        BNE   IS           ;; OF WORD
        MOV   (SP), R0         ;; 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)+, R0       ;; POP STACK INTO R0
        MOV   $PWRDN, @PWRVEC  ;; SET UP THE POWER DOWN VECTOR
        MOV   #340, @PWRVEC+2  ;; PRIO:7
        TYPE
        $PWRMG: .WORD $POWER    ;; REPORT THE POWER FAILURE
        MOV   (PC)+, (SP)      ;; POWER FAIL MESSAGE POINTER
        $PWRAD: .WORD START     ;; RESTART AT START
        RTI                    ;; RESTART ADDRESS
        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

```

ERROR MESSAGES, DATA HEADERS, DATA VECTORS, ETC

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

5399						
5400						
5401						; MESSAGE PREFIXES
5402	026034	047510	035124	000040	ASCHOT:	.ASCIZ "HOT: "
5403	026042	040527	046522	020072	ASCWRM:	.ASCIZ "WARM: "
5404	026050	000				
5405						
5406						; ERROR MESSAGES HERE
5407	026051	122	041505	044505	EMA:	.ASCIZ "RECEIVED FPS IS BAD"
5408	026056	042526	020104	050106		
5409	026064	020123	051511	041040		
5410	026072	042101	000			
5411	026075	122	041505	044505	EMB:	.ASCIZ "RECEIVED FEC/FEA IS BAD"
5412	026102	042526	020104	042506		
5413	026110	027503	042506	020101		
5414	026116	051511	041040	042101		
5415	026124	000				
5416	026125	101	051502	043050	EMC:	.ASCIZ "ABS(F/D) OPERATION - RESULT INCORRECT"
5417	026132	042057	020051	050117		
5418	026140	051105	052101	047511		
5419	026146	020116	020055	042522		
5420	026154	052523	052114	044440		
5421	026162	041516	051117	042522		
5422	026170	052103	000			
5423	026173	125	042516	050130	EMD:	.ASCIZ "UNEXPECTED FLOATING POINT TRAP, IGNORED & CONTINUING"
5424	026200	041505	042524	020104		
5425	026206	046106	040517	044524		
5426	026214	043516	050040	044517		
5427	026222	052116	052040	040522		
5428	026230	026120	044440	047107		
5429	026236	051117	042105	023040		
5430	026244	041440	047117	044524		
5431	026252	052516	047111	000107		
5432	026260	047125	041101	042514	EME:	.ASCIZ "UNABLE TO REFERENCE FLOATING ACCUMULATOR"
5433	026266	052040	020117	042522		
5434	026274	042506	042522	041516		
5435	026302	020105	046106	040517		
5436	026310	044524	043516	040440		
5437	026316	041503	046525	046125		
5438	026324	052101	051117	000		
5439	026331	122	050111	046120	EMF:	.ASCIZ "RIPPLING A 1 THRU FPS FAILED"
5440	026336	047111	020107	020101		
5441	026344	020061	044124	052522		
5442	026352	043040	051520	043040		
5443	026360	044501	042514	000104		
5444	026366	044522	050120	044514	EMG:	.ASCIZ "RIPPLING A 0 THRU FPS FAILED"
5445	026374	043516	040440	030040		
5446	026402	052040	051110	020125		
5447	026410	050106	020123	040506		
5448	026416	046111	042105	000		
5449	026423	123	052105	020106	EMH:	.ASCIZ "SETF FAILED TO CLEAR FPS BIT 7"
5450	026430	040506	046111	042105		
5451	026436	052040	020117	046103		
5452	026444	040505	020122	050106		
5453	026452	020123	044502	020124		
5454	026460	000067				

5455	026462	042523	042124	043040	EMI:	.ASCIZ	"SETD FAILED TO SET FPS BIT 7"
5456	026470	044501	042514	020104			
5457	026476	047524	051440	052105			
5458	026504	043040	051520	041040			
5459	026512	052111	033440	000			
5460	026517	123	052105	020111	EMJ:	.ASCIZ	"SETI FAILED TO CLEAR FPS BIT 6"
5461	026524	040506	046111	042105			
5462	026532	052040	020117	046103			
5463	026540	040505	020122	050106			
5464	026546	020123	044502	020124			
5465	026554	000066					
5466	026556	042523	046124	043040	EMK:	.ASCIZ	"SETL FAILED TO SET FPS BIT 6"
5467	026564	044501	042514	020104			
5468	026572	047524	051440	052105			
5469	026600	043040	051520	041040			
5470	026606	052111	033040	000			
5471	026613	124	051505	020124	EML:	.ASCII	"TEST OF SRC/DST ADDR MODE FAILED - "
5472	026620	043117	051440	041522			
5473	026626	042057	052123	040440			
5474	026634	042104	020122	047515			
5475	026642	042504	043040	044501			
5476	026650	042514	020104	020055			
5477	026656	042522	052523	052114		.ASCIZ	"RESULT DOESN'T CHECK"
5478	026664	042040	042517	047123			
5479	026672	052047	041440	042510			
5480	026700	045503	000				
5481	026703	124	051505	020124	EMM:	.ASCII	"TEST OF SRC/DST ADDR MODE FAILED - "
5482	026710	043117	051440	041522			
5483	026716	042057	052123	040440			
5484	026724	042104	020122	047515			
5485	026732	042504	043040	044501			
5486	026740	042514	020104	020055			
5487	026746	051123	020103	042101		.ASCIZ	"SRC ADDR REG WRONG INCRE/DECRE"
5488	026754	051104	051040	043505			
5489	026762	053440	047522	043516			
5490	026770	044440	041516	042522			
5491	026776	042057	041505	042522			
5492	027004	000					
5493	027005	124	051505	020124	EMN:	.ASCII	"TEST OF SRC/DST ADDR MODE FAILED - "
5494	027012	043117	051440	041522			
5495	027020	042057	052123	040440			
5496	027026	042104	020122	047515			
5497	027034	042504	043040	044501			
5498	027042	042514	020104	020055			
5499	027050	051504	020124	042101		.ASCIZ	"DST ADDR REG WRONG INCRE/DECRE"
5500	027056	051104	051040	043505			
5501	027064	053440	047522	043516			
5502	027072	044440	041516	042522			
5503	027100	042057	041505	042522			
5504	027106	000					
5505	027107	103	041506	020103	EMO:	.ASCIZ	"CFCC FAILED TO COPY CONDITION CODES CORRECTLY"
5506	027114	040506	046111	042105			
5507	027122	052040	020117	047503			
5508	027130	054520	041440	047117			
5509	027136	044504	044524	047117			
5510	027144	041440	042117	051505			

5511	027152	041440	051117	042522	
5512	027160	052103	054514	000	
5513	027165	124	051505	020124	EMP: .ASCII "TEST OF FSRC/FDST D ADDR MODE FAILED - "
5514	027172	043117	043040	051123	
5515	027200	027503	042106	052123	
5516	027206	042040	040440	042104	
5517	027214	020122	047515	042504	
5518	027222	043040	044501	042514	
5519	027230	020104	020055		
5520	027234	042522	052523	052114	.ASCIZ "RESULT DOESN'T CHECK"
5521	027242	042040	042517	047123	
5522	027250	052047	041440	042510	
5523	027256	045503	000		
5524	027261	124	051505	020124	EMQ: .ASCII "TEST OF FSRC/FDST D ADDR MODE FAILED - "
5525	027266	043117	043040	051123	
5526	027274	027503	042106	052123	
5527	027302	042040	040440	042104	
5528	027310	020122	047515	042504	
5529	027316	043040	044501	042514	
5530	027324	020104	020055		
5531	027330	051506	041522	040440	.ASCIZ "FSRC ADDR REG WRONG INCRE/DECRE"
5532	027336	042104	020122	042522	
5533	027344	020107	051127	047117	
5534	027352	020107	047111	051103	
5535	027360	027505	042504	051103	
5536	027366	000105			
5537	027370	042524	052123	047440	EMR: .ASCII "TEST OF FSRC/FDST D ADDR MODE FAILED - "
5538	027376	020106	051506	041522	
5539	027404	043057	051504	020124	
5540	027412	020104	042101	051104	
5541	027420	046440	042117	020105	
5542	027426	040506	046111	042105	
5543	027434	026440	040		
5544	027437	106	051504	020124	.ASCIZ "FDST ADDR REG WRONG INCRE/DECRE"
5545	027444	042101	051104	051040	
5546	027452	043505	053440	047522	
5547	027460	043516	044440	041516	
5548	027466	042522	042057	041505	
5549	027474	042522	000		
5550	027477	124	051505	020124	EMS: .ASCII "TEST OF FSRC/FDST F ADDR MODE FAILED - "
5551	027504	043117	043040	051123	
5552	027512	027503	042106	052123	
5553	027520	043040	040440	042104	
5554	027526	020122	047515	042504	
5555	027534	043040	044501	042514	
5556	027542	020104	020055		
5557	027546	042522	052523	052114	.ASCIZ "RESULT DOESN'T CHECK"
5558	027554	042040	042517	047123	
5559	027562	052047	041440	042510	
5560	027570	045503	000		
5561	027573	124	051505	020124	EMT: .ASCII "TEST OF FSRC/FDST F ADDR MODE FAILED - "
5562	027600	043117	043040	051123	
5563	027606	027503	042106	052123	
5564	027614	043040	040440	042104	
5565	027622	020122	047515	042504	
5566	027630	043040	044501	042514	

5567	027636	020104	020055		
5568	027642	051506	041522	040440	.ASCIZ "FSRC ADDR REG WRONG INCRE/DECRE"
5569	027650	042104	020122	042522	
5570	027656	020107	051127	047117	
5571	027664	020107	047111	051103	
5572	027672	027505	042504	051103	
5573	027700	000105			
5574	027702	042524	052123	047440	EMU: .ASCII "TEST OF FSRC/FDST F ADDR MODE FAILED - "
5575	027710	020106	051506	041522	
5576	027716	043057	051504	020124	
5577	027724	020106	042101	051104	
5578	027732	046440	042117	020105	
5579	027740	040506	046111	042105	
5580	027746	026440	040		
5581	027751	106	051504	020124	.ASCIZ "FDST ADDR REG WRONG INCRE/DECRE"
5582	027756	042101	051104	051040	
5583	027764	043505	053440	047522	
5584	027772	043516	044440	041516	
5585	030000	042522	042057	041505	
5586	030006	042522	000		
5587	030011	114	042104	051457	EMV: .ASCIZ "LDD/STD PATTERN IN FPP AC BAD"
5588	030016	042124	050040	052101	
5589	030024	042524	047122	044440	
5590	030032	020116	050106	020120	
5591	030040	041501	041040	042101	
5592	030046	000			
5593	030047	114	042104	046057	EMW: .ASCIZ "LDD/LDF/STD PATTERN IN FPP AC BAD"
5594	030054	043104	051457	042124	
5595	030062	050040	052101	042524	
5596	030070	047122	044440	020116	
5597	030076	050106	020120	041501	
5598	030104	041040	042101	000	
5599	030111	116	043505	043050	EMX: .ASCIZ "NEG(F/D) OPERATION - RESULT INCORRECT"
5600	030116	042057	020051	050117	
5601	030124	051105	052101	047511	
5602	030132	020116	020055	042522	
5603	030140	052523	052114	044440	
5604	030146	041516	051117	042522	
5605	030154	052103	000		
5606	030157	124	052123	043050	EMY: .ASCIZ "TST(F/D) OPERATION - OPERAND MODIFIED AFTER EXECUTION"
5607	030164	042057	020051	050117	
5608	030172	051105	052101	047511	
5609	030200	020116	020055	050117	
5610	030206	051105	047101	020104	
5611	030214	047515	044504	044506	
5612	030222	042105	040440	052106	
5613	030230	051105	042440	042530	
5614	030236	052503	044524	047117	
5615	030244	000			
5616	030245	103	051114	043050	EMZ: .ASCIZ "CLR(F/D) OPERATION - RESULT INCORRECT"
5617	030252	042057	020051	050117	
5618	030260	051105	052101	047511	
5619	030266	020116	020055	042522	
5620	030274	052523	052114	044440	
5621	030302	041516	051117	042522	
5622	030310	052103	000		

M11

FPU BASIC INSTR TESTS MACY11 27(1006) 25-APR-77 09:12 PAGE 123  
DQFPAB.P11 19-APR-77 13:36

ERROR MESSAGES, DATA HEADERS, DATA VECTORS, ETC

SEQ 0127

5623	030313	120	020103	047515
5624	030320	042504	031040	040440
5625	030326	042104	020122	020055
5626	030334	047111	051103	020105
5627	030342	043117	025440	000060
5628	030350	041520	046440	042117
5629	030356	020105	020062	042101
5630	030364	051104	026440	044440
5631	030372	041516	042522	047440
5632	030400	020106	032053	020054
5633	030406	033053	020054	051117
5634	030414	025440	030061	000
5635	030421	120	020103	047515
5636	030426	042504	031040	040440
5637	030434	042104	020122	020055
5638	030442	047111	051103	020105
5639	030450	043117	025440	020066
5640	030456	051117	025440	030061
5641	030464	000		
5642	030465	120	020103	047515
5643	030472	042504	031040	040440
5644	030500	042104	020122	020055
5645	030506	047111	051103	020105
5646	030514	043117	025440	030061
5647	030522	000		
5648				
5649				

EMAA: .ASCIZ "PC MODE 2 ADDR - INCRE OF +0"

EMAB: .ASCIZ "PC MODE 2 ADDR - INCRE OF +4, +6, OR +10"

EMAC: .ASCIZ "PC MODE 2 ADDR - INCRE OF +6 OR +10"

EMAD: .ASCIZ "PC MODE 2 ADDR - INCRE OF +10"

```

5650                                     ; DATA HEADERS HERE
5651 030523      114 040517 042504 DHA: .ASCIZ "LOADED STORED"
5652 030530      004504 052123 051117
5653 030536      042105      000
5654 030541      123 047524 042522 DHB: .ASCIZ "STORED"
5655 030546      000104
5656 030550      026455 026455 047514 DHC: .ASCIZ "----LOADED---- ----STORED----"
5657 030556      042101 042105 026455
5658 030564      026455 026411 026455
5659 030572      051455 047524 042522
5660 030600      026504 026455 000055
5661 030606      054105 023520 004504 DHD: .ASCIZ "EXP'D RCV'D"
5662 030614      041522 023526 000104
5663 030622      054105 023520 026504 DHE: .ASCIZ "EXP'D-FEC-RCV'D      EXP'D-FEA-RCV'D"
5664 030630      042506 026503 041522
5665 030636      023526 004504 054105
5666 030644      023520 026504 042506
5667 030652      026501 041522 023526
5668 030660      000104
5669 030662      026455 042455 050130 DHF: .ASCIZ "----EXPECTED--- ---RECEIVED---"
5670 030670      041505 042524 026504
5671 030676      026455 026411 026455
5672 030704      042522 042503 053111
5673 030712      042105 026455 000055
5674 030720      026455 026455 026455 DHG: .ASCII "-----EXPECTED-----"
5675 030726      026455 026455 042455
5676 030734      050130 041505 042524
5677 030742      026504 026455 026455
5678 030750      026455 026455 026455
5679 030756      011
5680 030757      055 026455 026455 .ASCIZ "-----RECEIVED-----"
5681 030764      026455 026455 026455
5682 030772      042522 042503 053111
5683 031000      042105 026455 026455
5684 031006      026455 026455 026455
5685 031014      000055
5686 031016      046117 020104 041520 DHH: .ASCIZ "OLD PC OLD PS NEW SP FPS FEC FEA"
5687 031024      047411 042114 050040
5688 031032      004523 042516 020127
5689 031040      050123 020011 050106
5690 031046      004523 043040 041505
5691 031054      020011 042506 000101
5692 031062      047514 042101 042105 DHI: .ASCIZ "LOADED STORED R2"
5693 031070      051411 047524 042522
5694 031076      004504 020040 031122
5695 031104      000
5696 031105      114 040517 042504 DHJ: .ASCIZ "LOADED STORED R3"
5697 031112      004504 052123 051117
5698 031120      042105 020011 051040
5699 031126      000063
5700 031130      047514 042101 042105 DHK: .ASCIZ "LOADED STORED R0 R2"
5701 031136      051411 047524 042522
5702 031144      004504 020040 030122
5703 031152      020011 051040 000062
5704 031160      047514 042101 042105 DHL: .ASCIZ "LOADED STORED $REG2 R0"
5705 031166      051411 047524 042522

```



5706	031174	004504	051044	043505					
5707	031202	004462	020040	030122					
5708	031210	000							
5709	031211	114	040517	042504	DHM:	.ASCIZ	"LOADED STORED \$REG2	R2"	
5710	031216	004504	052123	051117					
5711	031224	042105	022011	042522					
5712	031232	031107	020011	051040					
5713	031240	000062							
5714	031242	047514	042101	042105	DHM:	.ASCIZ	"LOADED STORED \$REG2	R4"	
5715	031250	051411	047524	042522					
5716	031256	004504	051044	043505					
5717	031264	004462	020040	032122					
5718	031272	000							
5719	031273	114	040517	042504	DHO:	.ASCIZ	"LOADED STORED \$REG2	R5"	
5720	031300	004504	052123	051117					
5721	031306	042105	022011	042522					
5722	031314	031107	020011	051040					
5723	031322	000065							
5724	031324	047514	042101	042105	DHP:	.ASCIZ	"LOADED STORED \$REG2	R1	R3"
5725	031332	051411	047524	042522					
5726	031340	004504	051044	043505					
5727	031346	004462	020040	030522					
5728	031354	020011	051040	000063					
5729	031362	047514	042101	042105	DHQ:	.ASCIZ	"LOADED STORED \$REG2	R3	R5"
5730	031370	051411	047524	042522					
5731	031376	004504	051044	043505					
5732	031404	004462	020040	031522					
5733	031412	020011	051040	000065					
5734	031420	026455	026455	026455	DHR:	.ASCIZ	"-----STORED-----"		
5735	031426	026455	026455	026455					
5736	031434	052123	051117	042105					
5737	031442	026455	026455	026455					
5738	031450	026455	026455	026455					
5739	031456	000							
5740	031457	055	026455	026455	DHS:	.ASCIZ	"-----STORED-----"	R4"	
5741	031464	026455	026455	026455					
5742	031472	051455	047524	042522					
5743	031500	026504	026455	026455					
5744	031506	026455	026455	026455					
5745	031514	004455	020040	032122					
5746	031522	000							
5747	031523	055	026455	026455	DHT:	.ASCIZ	"-----STORED-----"	R5"	
5748	031530	026455	026455	026455					
5749	031536	051455	047524	042522					
5750	031544	026504	026455	026455					
5751	031552	026455	026455	026455					
5752	031560	004455	020040	032522					
5753	031566	000							
5754	031567	055	026455	026455	DHU:	.ASCII	"-----STORED-----"	R0"	
5755	031574	026455	026455	026455					
5756	031602	051455	047524	042522					
5757	031610	026504	026455	026455					
5758	031616	026455	026455	026455					
5759	031624	004455	020040	030122					
5760	031632	020011	051040	000063	DHV:	.ASCIZ	"-----R3-----"		
5761	031640	026455	026455	026455		.ASCII	"-----STORED-----"	R1"	

5762	031646	026455	026455	026455					
5763	031654	052123	051117	042105					
5764	031662	026455	026455	026455					
5765	031670	026455	026455	026455					
5766	031676	020011	051040	061					
5767	031703	040	004440	031522		.ASCIZ	"	R3"	
5768	031710	000							
5769	031711	055	026455	026455	DHW:	.ASCII	"-----STORED-----	\$REG10"	
5770	031716	026455	026455	026455					
5771	031724	051455	047524	042522					
5772	031732	026504	026455	026455					
5773	031740	026455	026455	026455					
5774	031746	004455	051044	043505					
5775	031754	030061							
5776	031756	020040	051011	000060		.ASCIZ	"	R0"	
5777	031764	026455	026455	026455	DHX:	.ASCII	"-----STORED-----	\$REG10"	
5778	031772	026455	026455	026455					
5779	032000	052123	051117	042105					
5780	032006	026455	026455	026455					
5781	032014	026455	026455	026455					
5782	032022	022011	042522	030507					
5783	032030	060							
5784	032031	040	004440	031122		.ASCIZ	"	R2"	
5785	032036	000							
5786	032037	055	026455	026455	DHY:	.ASCII	"-----STORED-----	\$REG10"	
5787	032044	026455	026455	026455					
5788	032052	051455	047524	042522					
5789	032060	026504	026455	026455					
5790	032066	026455	026455	026455					
5791	032074	004455	051044	043505					
5792	032102	030061							
5793	032104	020040	051011	000064		.ASCIZ	"	R4"	
5794	032112	026455	026455	026455	DHZ:	.ASCII	"-----STORED-----	\$REG10"	
5795	032120	026455	026455	026455					
5796	032126	052123	051117	042105					
5797	032134	026455	026455	026455					
5798	032142	026455	026455	026455					
5799	032150	022011	042522	030507					
5800	032156	060							
5801	032157	011	051044	043505		.ASCIZ	"	\$REG11 R1"	
5802	032164	030461	020011	051040					
5803	032172	000061							
5804	032174	026455	026455	026455	DHAA:	.ASCII	"-----STORED-----	\$REG10"	
5805	032202	026455	026455	026455					
5806	032210	052123	051117	042105					
5807	032216	026455	026455	026455					
5808	032224	026455	026455	026455					
5809	032232	022011	042522	030507					
5810	032240	060							
5811	032241	011	051044	043505		.ASCIZ	"	\$REG11 R4 R5"	
5812	032246	030461	020011	051040					
5813	032254	004464	020040	032522					
5814	032262	000							
5815	032263	055	026455	046055	DHAB:	.ASCIZ	"-----LOADED-----	-----STORED-----	R1"
5816	032270	040517	042504	026504					
5817	032276	026455	004455	026455					

5818	032304	026455	052123	051117				
5819	032312	042105	026455	026455				
5820	032320	020011	051040	000061				
5821	032326	026455	026455	047514	DHAC:	.ASCIZ	"----LOADED----	----STORED---- R3"
5822	032334	042101	042105	026455				
5823	032342	026455	026411	026455				
5824	032350	051455	047524	042522				
5825	032356	026504	026455	004455				
5826	032364	020040	031522	000				
5827	032371	055	026455	046055	DHAD:	.ASCII	"----LOADED----	----STORED---- R2"
5828	032376	040517	042504	026504				
5829	032404	026455	004455	026455				
5830	032412	026455	052123	051117				
5831	032420	042105	026455	026455				
5832	032426	020011	051040	062				
5833	032433	011	020040	032522		.ASCIZ	"	R5"
5834	032440	000						
5835	032441	055	026455	046055	DHAE:	.ASCIZ	"----LOADED----	----STORED---- \$REG4"
5836	032446	040517	042504	026504				
5837	032454	026455	004455	026455				
5838	032462	026455	052123	051117				
5839	032470	042105	026455	026455				
5840	032476	022011	042522	032107				
5841	032504	000						
5842	032505	055	026455	046055	DHAF:	.ASCII	"----LOADED----	----STORED---- \$REG4"
5843	032512	040517	042504	026504				
5844	032520	026455	004455	026455				
5845	032526	026455	052123	051117				
5846	032534	042105	026455	026455				
5847	032542	022011	042522	032107				
5848	032550	020011	051040	004460		.ASCIZ	"	R0 R4"
5849	032556	020040	032122	000				
5850	032563	055	026455	046055	DHAG:	.ASCII	"----LOADED----	----STORED---- \$REG4"
5851	032570	040517	042504	026504				
5852	032576	026455	004455	026455				
5853	032604	026455	052123	051117				
5854	032612	042105	026455	026455				
5855	032620	022011	042522	032107				
5856	032626	020011	051040	004461		.ASCIZ	"	R1 R2"
5857	032634	020040	031122	000				
5858	032641	055	026455	046055	DHAH:	.ASCII	"----LOADED----	----STORED---- \$REG4"
5859	032646	040517	042504	026504				
5860	032654	026455	004455	026455				
5861	032662	026455	052123	051117				
5862	032670	042105	026455	026455				
5863	032676	022011	042522	032107				
5864	032704	020011	051040	004464		.ASCIZ	"	R4 R5"
5865	032712	020040	032522	000				
5866	032717	055	026455	046055	DHAI:	.ASCII	"----LOADED----	----STORED---- \$REG4"
5867	032724	040517	042504	026504				
5868	032732	026455	004455	026455				
5869	032740	026455	052123	051117				
5870	032746	042105	026455	026455				
5871	032754	022011	042522	032107				
5872	032762	022011	042522	032507		.ASCIZ	"	\$REG5 R5"
5873	032770	020011	051040	000065				

5874	032776	026455	026455	047514
5875	033004	042101	042105	026455
5876	033012	026455	026411	026455
5877	033020	051455	047524	042522
5878	033026	026504	026455	004455
5879	033034	051044	043505	064
5880	033041	011	051044	043505
5881	033046	004465	020040	030122
5882	033054	020011	051040	000062
5883	033062	026455	026455	047514
5884	033070	042101	042105	026455
5885	033076	026455	026411	026455
5886	033104	051455	047524	042522
5887	033112	026504	026455	000055
5888	033120	000040		
5889				
5890				

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

.ASCIZ " \$REG5 R0 R2"

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

DHAL: .ASCIZ " "

					; DATA ADDRESS VECTOR	
5891						.EVEN
5892						.WORD
5893	033122	001170	000000		DTA:	.WORD \$REG0,0
5894	033126	001170	001172	000000	DTB:	.WORD \$REG0,\$REG1,0
5895	033134	001170	001172	001174	DTC:	.WORD \$REG0,\$REG1,\$REG2,\$REG3,0
5896	033142	001176	000000			
5897	033146	001242	002330	000000	DTD:	.WORD \$TMP5,FPS,0
5898	033154	001252	002330	000000	DTE:	.WORD \$TMP11,FPS,0
5899	033162	001256	002330	000000	DTF:	.WORD \$TMP13,FPS,0
5900	033170	001244	002332	002344	DTG:	.WORD \$TMP6,FEC,EXPFEA,FEA,0
5901	033176	002334	000000			
5902	033202	001254	002332	002344	DTH:	.WORD \$TMP12,FEC,EXPFEA,FEA,0
5903	033210	002334	000000			
5904	033214	001260	002332	002344	DTI:	.WORD \$TMP14,FEC,EXPFEA,FEA,0
5905	033222	002334	000000			
5906	033226	001234	001236	001170	DTJ:	.WORD \$TMP2,\$TMP3,\$REG0,\$REG1,0
5907	033234	001172	000000			
5908	033240	001240	001242	001244	DTK:	.WORD \$TMP4,\$TMP5,\$TMP6,\$TMP7
5909	033246	001246				
5910	033250	001170	001172	001174		.WORD \$REG0,\$REG1,\$REG2,\$REG3,0
5911	033256	001176	000000			
5912	033262	001244	001246	001250	DTL:	.WORD \$TMP6,\$TMP7,\$TMP10,\$TMP11
5913	033270	001252				
5914	033272	001170	001172	001174		.WORD \$REG0,\$REG1,\$REG2,\$REG3,0
5915	033300	001176	000000			
5916	033304	002336	002340	002342	DTM:	.WORD FPPOPC,FPPOPS,FPPOSP,FPS,FEC,FEA,0
5917	033312	002330	002332	002334		
5918	033320	000000				
5919	033322	001170	001172	002352	DTN:	.WORD \$REG0,\$REG1,EREG2,0
5920	033330	000000				
5921	033332	001170	001172	002354	DTO:	.WORD \$REG0,\$REG1,EREG3,0
5922	033340	000000				
5923	033342	001170	001172	002346	DTP:	.WORD \$REG0,\$REG1,EREG0,EREG2,0
5924	033350	002352	000000			
5925	033354	001170	001172	001174	DTQ:	.WORD \$REG0,\$REG1,\$REG2,EREG0,0
5926	033362	002346	000000			
5927	033366	001170	001172	001174	DTR:	.WORD \$REG0,\$REG1,\$REG2,EREG2,0
5928	033374	002352	000000			
5929	033400	001170	001172	001174	DTS:	.WORD \$REG0,\$REG1,\$REG2,EREG4,0
5930	033406	002356	000000			
5931	033412	001170	001172	001174	DTT:	.WORD \$REG0,\$REG1,\$REG2,EREG5,0
5932	033420	002360	000000			
5933	033424	001170	001172	001174	DTU:	.WORD \$REG0,\$REG1,\$REG2,EREG1,EREG3,0
5934	033432	002350	002354	000000		
5935	033440	001170	001172	001174	DTV:	.WORD \$REG0,\$REG1,\$REG2,EREG3,EREG5,0
5936	033446	002354	002360	000000		
5937	033454	001200	001202	001204	DTW:	.WORD \$REG4,\$REG5,\$REG6,\$REG7,0
5938	033462	001206	000000			
5939	033466	001200	001202	001204	DTX:	.WORD \$REG4,\$REG5,\$REG6,\$REG7,EREG4,0
5940	033474	001206	002356	000000		
5941	033502	001200	001202	001204	DTY:	.WORD \$REG4,\$REG5,\$REG6,\$REG7,EREG5,0
5942	033510	001206	002360	000000		
5943	033516	001200	001202	001204	DTZ:	.WORD \$REG4,\$REG5,\$REG6,\$REG7,EREG0,EREG3,0
5944	033524	001206	002346	002354		
5945	033532	000000				
5946	033534	001200	001202	001204	DTAA:	.WORD \$REG4,\$REG5,\$REG6,\$REG7,EREG1,EREG3,0

5947	033542	001206	002350	002354			
5948	033550	000000					
5949	033552	001200	001202	001204	DTAB:	.WORD	\$REG4, \$REG5, \$REG6, \$REG7, \$REG10, EREG0, 0
5950	033560	001206	001210	002346			
5951	033566	000000					
5952	033570	001200	001202	001204	DTAC:	.WORD	\$REG4, \$REG5, \$REG6, \$REG7, \$REG10, EREG2, 0
5953	033576	001206	001210	002352			
5954	033604	000000					
5955	033606	001200	001202	001204	DTAD:	.WORD	\$REG4, \$REG5, \$REG6, \$REG7, \$REG10, EREG4, 0
5956	033614	001206	001210	002356			
5957	033622	000000					
5958	033624	001200	001202	001204	DTAE:	.WORD	\$REG4, \$REG5, \$REG6, \$REG7
5959	033632	001206					
5960	033634	001210	001212	002350		.WORD	\$REG10, \$REG11, EREG1, 0
5961	033642	000000					
5962	033644	001200	001202	001204	DTAF:	.WORD	\$REG4, \$REG5, \$REG6, \$REG7
5963	033652	001206					
5964	033654	001210	001212	002356		.WORD	\$REG10, \$REG11, EREG4, EREG5, 0
5965	033662	002360	000000				
5966	033666	001170	001172	001174	DTAG:	.WORD	\$REG0, \$REG1, \$REG2, \$REG3, EREG1, 0
5967	033674	001176	002350	000000			
5968	033702	001170	001172	001174	DTAH:	.WORD	\$REG0, \$REG1, \$REG2, \$REG3, EREG3, 0
5969	033710	001176	002354	000000			
5970	033716	001170	001172	001174	DTAI:	.WORD	\$REG0, \$REG1, \$REG2, \$REG3, EREG2, EREG5, 0
5971	033724	001176	002352	002360			
5972	033732	000000					
5973	033734	001170	001172	001174	DTAJ:	.WORD	\$REG0, \$REG1, \$REG2, \$REG3, \$REG4, 0
5974	033742	001176	001200	000000			
5975	033750	001170	001172	001174	DTAK:	.WORD	\$REG0, \$REG1, \$REG2, \$REG3
5976	033756	001176					
5977	033760	001200	002346	002356		.WORD	\$REG4, EREG0, EREG4, 0
5978	033766	000000					
5979	033770	001170	001172	001174	DTAL:	.WORD	\$REG0, \$REG1, \$REG2, \$REG3
5980	033776	001176					
5981	034000	001200	002350	002352		.WORD	\$REG4, EREG1, EREG2, 0
5982	034006	000000					
5983	034010	001170	001172	001174	DTAM:	.WORD	\$REG0, \$REG1, \$REG2, \$REG3
5984	034016	001176					
5985	034020	001200	002356	002360		.WORD	\$REG4, EREG4, EREG5, 0
5986	034026	000000					
5987	034030	001170	001172	001174	DTAN:	.WORD	\$REG0, \$REG1, \$REG2, \$REG3
5988	034036	001176					
5989	034040	001200	001202	002360		.WORD	\$REG4, \$REG5, EREG5, 0
5990	034046	000000					
5991	034050	001170	001172	001174	DTAO:	.WORD	\$REG0, \$REG1, \$REG2, \$REG3
5992	034056	001176					
5993	034060	001200	001202	002346		.WORD	\$REG4, \$REG5, EREG0, EREG2, 0
5994	034066	002352	000000				
5995	034072	000000			DTAP:	.WORD	0
5996							
5997							
5998							
5999	000001						

; THE END  
.END









CLRF2	016704	3699	3704#					
CLRF3	016736	3716	3721#					
CLRF4	016770	3733	3738#					
CLRF5	017022	3750	3755#					
CLRF6	017054	3767	3772#					
CR	= 000015	41#	486	491	499	5171	5181	
CRLF	= 000200	42#	5142	5181				
DOISP	= 177570	48#	263	536				
DHA	030523	368	369	393	5651#			
DHAA	032174	398	399	5804#				
DHAB	032263	431	5815#					
DHAC	032326	422	423	425	5821#			
DHAD	032371	414	415	5827#				
DHAE	032441	429	5835#					
DHAF	032505	416	417	5842#				
DHAG	032563	421	5850#					
DHAH	032641	426	427	428	5858#			
DHAI	032717	424	5866#					
DHAJ	032776	418	419	420	5874#			
DHAK	033062	430	5883#					
DHAL	033120	456	457	458	459	5888#		
DHB	030541	370	371	372	373	5654#		
DHC	030550	433	5656#					
DHD	030606	435	436	437	5661#			
DHE	030622	439	440	441	5663#			
DHF	030662	445	447	449	451	5669#		
DHG	030720	443	444	446	448	450	452	5674#
DHH	031016	454	5686#					
DHI	031062	390	5692#					
DHJ	031105	388	389	5696#				
DHK	031130	383	384	5700#				
DHL	031160	381	382	5704#				
DHM	031211	387	5709#					
DHN	031242	385	386	391	5714#			
DHO	031273	375	5719#					
DHP	031324	378	379	380	5724#			
DHQ	031362	376	377	5729#				
DHR	031420	411	5734#					
DHS	031457	396	397	5740#				
DHT	031523	395	400	5747#				
DHU	031567	401	402	5754#				
DHV	031640	409	410	5761#				
DHW	031711	405	5769#					
DHX	031764	407	408	412	5777#			
DHY	032037	406	5786#					
DHZ	032112	403	404	5794#				
DISPLA	001146	263#	536#	544#	4969#	4999#		
DISPRE	000174	194#	544					
DSMR	= 177570	47#	262	535				
DTA	033122	370	371	372	373	5893#		
DTAA	033534	409	410	5946#				
DTAB	033552	405	5949#					
DTAC	033570	407	408	412	5952#			
DTAD	033606	406	5955#					
DTAE	033624	403	404	5958#				
DTAF	033644	398	399	5962#				















TSTF4	017540	3951	3956#
TSTF5	017572	3968	3973#
TSTF6	017624	3985	3990#
TSTF7	017656	4002	4007#
TST1	003576	638#	
TST10	004144	757	765#
TST100	007232	1701	1708#
TST101	007252	1713	1720#
TST102	007272	1725	1733#
TST103	007312	1738	1745#
TST104	007332	1750	1757#
TST105	007352	1762	1769#
TST106	007372	1774	1781#
TST107	007412	1786	1793#
TST11	004224	785	793#
TST110	007432	1798	1806#
TST111	007452	1811	1818#
TST112	007472	1823	1830#
TST113	007512	1835	1842#
TST114	007532	1847	1854#
TST115	007552	1859	1866#
TST116	007572	1871	1879#
TST117	007612	1884	1891#
TST12	004314	818	826#
TST120	007632	1896	1903#
TST121	007652	1908	1915#
TST122	007672	1920	1927#
TST123	007712	1932	1940#
TST124	007732	1945	1952#
TST125	007752	1957	1964#
TST126	007772	1969	1976#
TST127	010012	1981	1988#
TST13	004372	845	853#
TST130	010032	1993	2007#
TST131	010156	2030	2039#
TST132	010312	2067	2076#
TST133	010466	2107	2116#
TST134	010622	2144	2153#
TST135	010760	2182	2191#
TST136	011130	2221	2230#
TST137	011302	2260	2269#
TST14	004444	872	880#
TST140	011436	2293	2302#
TST141	011574	2332	2341#
TST142	011734	2370	2379#
TST143	012060	2402	2411#
TST144	012224	2441	2450#
TST145	012322	2473	2481#
TST146	012430	2505	2513#
TST147	012552	2543	2551#
TST15	004522	899	907#
TST150	012650	2570	2578#
TST151	012744	2600	2608#
TST152	013046	2627	2635#
TST153	013142	2657	2665#
TST154	013256	2694	2702#

FPU BASIC INSTR TESTS MACY11 27(1006)  
 DGFPA8.P11 19-APR-77 13:36

25-APR-77 09:12 PAGE 143  
 CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0146

TST155	013364	2725	2733#
TST156	013446	2749	2757#
TST157	013546	2779	2787#
TST16	004576	922	930#
TST160	013636	2805	2818#
TST161	014166	2885#	
TST162	014220	2889	2902#
TST163	014252	2906	2919#
TST164	014304	2923	2936#
TST165	014336	2940	2953#
TST166	014370	2957	2970#
TST167	014422	2974	2987#
TST17	004672	951	959#
TST170	014454	2991	3004#
TST171	014506	3008	3021#
TST172	014540	3025	3038#
TST173	014572	3042	3055#
TST174	014624	3059	3073#
TST175	014666	3077	3092#
TST176	014730	3096	3111#
TST177	014772	3115	3130#
TST2	003662	660#	
TST20	004760	983	991#
TST200	015034	3134	3149#
TST201	015076	3153	3168#
TST202	015140	3172	3187#
TST203	015202	3191	3206#
TST204	015244	3210	3225#
TST205	015306	3229	3244#
TST206	015350	3248	3263#
TST207	015412	3267	3283#
TST21	005022	1004	1012#
TST210	015444	3287	3300#
TST211	015476	3304	3317#
TST212	015530	3321	3334#
TST213	015562	3338	3351#
TST214	015614	3355	3368#
TST215	015646	3372	3385#
TST216	015700	3389	3402#
TST217	015732	3406	3419#
TST22	005062	1025	1033#
TST220	015764	3423	3436#
TST221	016016	3440	3453#
TST222	016050	3457	3471#
TST223	016112	3475	3490#
TST224	016154	3494	3509#
TST225	016216	3513	3528#
TST226	016260	3532	3547#
TST227	016322	3551	3566#
TST23	005140	1052	1065#
TST230	016364	3570	3585#
TST231	016426	3589	3604#
TST232	016470	3608	3623#
TST233	016532	3627	3642#
TST234	016574	3646	3661#
TST235	016636	3665	3681#

TST236	016670	3685	3698#
TST237	016722	3702	3715#
TST24	005272	1086	1095#
TST240	016754	3719	3732#
TST241	017006	3736	3749#
TST242	017040	3753	3766#
TST243	017072	3770	3784#
TST244	017134	3788	3803#
TST245	017176	3807	3822#
TST246	017240	3826	3841#
TST247	017302	3845	3860#
TST25	005340	1100	1113#
TST250	017344	3864	3879#
TST251	017406	3883	3899#
TST252	017440	3903	3916#
TST253	017472	3920	3933#
TST254	017524	3937	3950#
TST255	017556	3954	3967#
TST256	017610	3971	3984#
TST257	017642	3988	4001#
TST26	005406	1118	1131#
TST260	017674	4005	4018#
TST261	017726	4022	4035#
TST262	017760	4039	4052#
TST263	020012	4056	4069#
TST264	020044	4073	4087#
TST265	020106	4091	4106#
TST266	020150	4110	4125#
TST267	020212	4129	4144#
TST27	005454	1136	1149#
TST270	020254	4148	4163#
TST271	020316	4167	4182#
TST272	020360	4186	4201#
TST273	020422	4205	4220#
TST274	020464	4224	4239#
TST275	020526	4243	4258#
TST276	020570	4262	4277#
TST277	020632	4281	4296#
TST3	003746	678#	
TST30	005522	1154	1167#
TST31	005570	1172	1185#
TST32	005636	1190	1203#
TST33	005656	1208	1215#
TST34	005676	1220	1227#
TST35	005716	1232	1239#
TST36	005736	1244	1251#
TST37	005756	1256	1263#
TST4	003774	694#	
TST40	005776	1268	1276#
TST41	006016	1281	1288#
TST42	006036	1293	1300#
TST43	006056	1305	1312#
TST44	006076	1317	1324#
TST45	006116	1329	1336#
TST46	006136	1341	1349#
TST47	006156	1354	1361#



SCBLF = 001321	319#	603	5010	5034	5050	5070	5076	5093	5146	5181				
SECTR = 000006	1200#	1212#	1224#	1236#	1248#	1260#	1272#	1273#	1285#	1297#	1309#	1321#	1333#	
	1345#	1346#	1358#	1370#	1382#	1394#	1406#	1418#	1419#	1431#	1443#	1455#	1467#	
	1479#	1480#	1492#	1504#	1516#	1528#	1540#	1657#	1669#	1681#	1693#	1705#	1717#	
	1729#	1730#	1742#	1754#	1766#	1778#	1790#	1802#	1803#	1815#	1827#	1839#	1851#	
	1863#	1875#	1876#	1888#	1900#	1912#	1924#	1936#	1937#	1949#	1961#	1973#	1985#	
	1997#													
SDEVCT 001334	331#													
SDOAGN 021026	4351	4360	4368#											
SENDAD 021016	205	4363#	5029											
SENDOCT 020766	525	4353#												
SENV 001344	336#	5012	5125	5193	5217									
SENVN 001345	337#	548	5127	5132	5195									
SEOP 020730	4314	4317	4329	4343#										
SEOPCT 020760	525*	4350#	4354											
SERFLG 001104	244#	4299*	4344*	4943	4945	4951*	4973	4997*	5034					
SERMAX 001120	250#	528*	4945	4968*	4973									
SERROR 024142	519	4987#												
SERRPC 001122	251#	5004*	5005*	5006	5034	5057	5096							
SERRTB 001354	351#	5065												
SERTTL 001114	248#	5003*	5034											
SESCAP 001312	316#	527*	4967*	5025	5027	5034								
SETABL 001344	335#													
SETEND 001354	231	347#												
SFATAL 001326	328#	5221*												
SFFLG 025346	5184*	5187*	5215	5224*	5232#									
SFILLC 001164	271#	5150	5181											
SFILLS 001163	270#	5181												
SGADR 001124	252#													
SGDAT 001130	254#													
SGET42 021006	4356	4359#												
SGTSMR= ***** U	5351													
SHD = 000000	12													
SHIBTS 001000	226#													
SICNT 001106	245#	4958*	4959	4961*	4972									
SILLUP 026016	5356	5372	5393#											
SINTAG 001141	259#													
SITEMB 001116	249#	5006*	5014	5034	5054									
SLF 001322	320#	5034	5181											
SLFLG 025345	5225*	5231#												
SLPADR 001110	246#	529*	4949*	4965*	4970	4972								
SLPERR 001112	247#	530*	640*	662*	1068*	4379*	4454*	4542*	4585*	4633*	4676*	4724*	4767*	
	4816#	4859*	4949	4966*	4972	5024								
	264#	4941												
SLPTST 001150	227	231	326#	547	4964	5012	5125							
SMAIL 001324	227#													
SMBADR 001002	227#													
SMFLG 025344	5185*	5191	5226*	5230#										
SMSGAD 001340	333#	5201*	5204											
SMSGLG 001342	334#	5206*												
SMSGTY 001324	327#	5199	5207*	5219	5223*									
SMXCNT 024140	4962	4972#												
SNLL 001162	269#	5152	5181											
SNWTST= 000001	635#	657#	675#	691#	707#	723#	740#	762#	790#	823#	850#	877#	904#	
	927#	956#	988#	1009#	1030#	1062#	1092#	1110#	1128#	1146#	1164#	1182#	1200#	
	1212#	1224#	1236#	1248#	1260#	1273#	1285#	1297#	1309#	1321#	1333#	1346#	1358#	
	1370#	1382#	1394#	1406#	1419#	1431#	1443#	1455#	1467#	1480#	1492#	1504#	1516#	

1528#	1542#	1561#	1580#	1599#	1618#	1637#	1657#	1669#	1681#	1693#	1705#	1717#
1730#	1742#	1754#	1766#	1778#	1790#	1803#	1815#	1827#	1839#	1851#	1863#	1876#
1888#	1900#	1912#	1924#	1937#	1949#	1961#	1973#	1985#	2004#	2036#	2073#	2113#
2150#	2188#	2227#	2266#	2299#	2338#	2376#	2408#	2447#	2478#	2510#	2548#	2575#
2605#	2632#	2662#	2699#	2730#	2754#	2784#	2815#	2881#	2883#	2898#	2900#	2915#
2917#	2932#	2934#	2949#	2951#	2966#	2968#	2983#	2985#	3000#	3002#	3017#	3019#
3034#	3036#	3051#	3053#	3069#	3071#	3088#	3090#	3107#	3109#	3126#	3128#	3145#
3147#	3164#	3166#	3183#	3185#	3202#	3204#	3221#	3223#	3240#	3242#	3259#	3261#
3279#	3281#	3296#	3298#	3313#	3315#	3330#	3332#	3347#	3349#	3364#	3366#	3381#
3383#	3398#	3400#	3415#	3417#	3432#	3434#	3449#	3451#	3467#	3469#	3486#	3488#
3505#	3507#	3524#	3526#	3543#	3545#	3562#	3564#	3581#	3583#	3600#	3602#	3619#
3621#	3638#	3640#	3657#	3659#	3677#	3679#	3694#	3696#	3711#	3713#	3728#	3730#
3745#	3747#	3762#	3764#	3780#	3782#	3799#	3801#	3818#	3820#	3837#	3839#	3856#
3858#	3875#	3877#	3895#	3897#	3912#	3914#	3929#	3931#	3946#	3948#	3963#	3965#
3980#	3982#	3997#	3999#	4014#	4016#	4031#	4033#	4048#	4050#	4065#	4067#	4083#
4085#	4102#	4104#	4121#	4123#	4140#	4142#	4159#	4161#	4178#	4180#	4197#	4199#
4216#	4218#	4235#	4237#	4254#	4256#	4273#	4275#					
5270#	5299#	5312#										
5265#	5269#	5274#	5277#	5288#	5314#							
4926#	4942#	4950#	4960#	4969#								
330#	547#	599#	4347#	4348#	4956#	4973#						
229#												
5389#	5396#											
5391#												
523#	5356#	5386#										
5389#												
5366#	5372#											
318#	5034#	5181#										
5352#												
5352#												
5352#												
5352#												
273#												
275#	642#	643#	644#	645#	647#	664#	665#	666#	667#	669#	681#	682#
697#	698#	713#	714#	729#	730#	744#	746#	752#	766#	768#	780#	794#
796#	803#	813#	827#	829#	840#	854#	856#	862#	867#	881#	883#	894#
908#	910#	917#	931#	933#	946#	960#	962#	978#	992#	994#	999#	1013#
1015#	1020#	1034#	1036#	1047#	1070#	1071#	1072#	1081#	2009#	2014#	2019#	2041#
2046#	2051#	2056#	2078#	2083#	2096#	2118#	2123#	2133#	2155#	2160#	2166#	2171#
2193#	2198#	2210#	2232#	2237#	2249#	2271#	2276#	2282#	2304#	2309#	2327#	2343#
2348#	2359#	2381#	2386#	2391#	2413#	2418#	2430#	2452#	2455#	2466#	2483#	2486#
2498#	2515#	2518#	2536#	2553#	2556#	2563#	2580#	2583#	2593#	2610#	2613#	2620#
2637#	2640#	2645#	2650#	2667#	2670#	2677#	2687#	2704#	2707#	2718#	2735#	2738#
2742#	2759#	2762#	2774#	2789#	2792#	2798#	2820#	2822#	2825#	2828#	2830#	2832#
2834#	2835#	4402#	4419#	4423#	4427#	4431#	4437#	4443#	4477#	4496#	4502#	4508#
4514#	4523#	4532#	4545#	4549#	4567#	4588#	4594#	4612#	4636#	4640#	4658#	4679#
4685#	4703#	4727#	4731#	4749#	4770#	4776#	4794#	4819#	4823#	4841#	4852#	4868#
4886#	5893#	5894#	5895#	5906#	5910#	5914#	5919#	5921#	5923#	5925#	5927#	5929#
5931#	5933#	5935#	5966#	5968#	5970#	5973#	5975#	5979#	5983#	5987#	5991#	5999#
276#	646#	647#	668#	669#	749#	752#	771#	780#	799#	813#	833#	840#
860#	867#	887#	894#	913#	917#	944#	946#	970#	973#	978#	996#	999#
1017#	1020#	1039#	1042#	1047#	1079#	1080#	1081#	2010#	2021#	2042#	2058#	2079#
2098#	2119#	2135#	2156#	2173#	2194#	2212#	2233#	2251#	2272#	2284#	2305#	2344#
2361#	2382#	2393#	2414#	2432#	2453#	2468#	2484#	2500#	2516#	2538#	2554#	2565#
2581#	2595#	2611#	2622#	2638#	2652#	2668#	2689#	2705#	2720#	2736#	2744#	2760#
2790#	2800#	2821#	2824#	2827#	2829#	2831#	2833#	2836#	2840#	2843#	2847#	2850#

SOCNT 025572  
SOMODE 025574  
SOVER 024124  
SPASS 001332  
SPASTM 001006  
SPOWER 026024  
SPWRAD 026012  
SPWRDN 025644  
SPWRMG 026006  
SPWRUP 025716  
SQUES 001320  
SROCHR= \*\*\*\*\*  
SRODEC= \*\*\*\*\*  
SRDLIN= \*\*\*\*\*  
SRODOCT= \*\*\*\*\*  
SREGAD 001166  
SREGO 001170

U  
U  
U  
U

SREG1 001172

		2854	2857*	2861	2864*	2869	2872*	2877	4404	4479	4546*	4569	4589*	4614
		4637*	4660	4680*	4705	4728*	4751	4771*	4796	4820*	4843	4863*	4888	5894
		5895	5906	5910	5914	5919	5921	5923	5925	5927	5929	5931	5933	5935
		5966	5968	5970	5973	5975	5979	5983	5987	5991				
\$REG10	001210	283#	2083*	2084	2198*	2199	2205	2245*	2246	2276*	2277	2309*	2310	2322
		2421*	2422	2425	5949	5952	5955	5960	5964					
\$REG11	001212	284#	2087*	2088	2091	2202*	2203*	5960	5964					
\$REG12	001214	285#												
\$REG13	001216	286#												
\$REG14	001220	287#												
\$REG15	001222	288#												
\$REG16	001224	289#												
\$REG17	001226	290#												
\$REG2	001174	277#	749*	750*	771*	772	775	799*	800	808	829*	830	835	883*
		884	889	913*	914	933*	934	1036*	1037	2011*	2023	2043*	2060	2080*
		2100	2120*	2137	2157*	2175	2195*	2214	2234*	2253	2273*	2286	2306*	2345*
		2363	2383*	2395	2415*	2434	2458	2461	2466	2490	2498	2522	2536	2560
		2563	2585	2588	2593	2616	2620	2643*	2650	2673	2687	2715	2718	2740*
		2742	2772*	2774	2796*	2798	2837*	2838	2844*	2845	2851*	2852	2858*	2859
		2866*	2867	2874*	2875	4406	4481	4590*	4616	4681*	4707	4772*	4798	4864*
		4890	5895	5910	5914	5925	5927	5929	5931	5933	5935	5966	5968	5970
		5973	5975	5979	5983	5987	5991							
\$REG3	001176	278#	2012*	2025	2044*	2062	2081*	2102	2121*	2139	2158*	2177	2196*	2216
		2235*	2255	2274*	2288	2307*	2346*	2365	2384*	2397	2416*	2436	2468	2500
		2538	2565	2595	2622	2652	2689	2720	2744	2800	2840	2847	2854	2861
		2869	2877	4408	4483	4591*	4618	4682*	4709	4773*	4800	4865*	4892	5895
		5910	5914	5966	5968	5970	5973	5975	5979	5983	5987	5991		
\$REG4	001200	279#	2017*	2019	2049*	2056	2087	2096	2125	2128	2133	2163	2171	2202
		2210	2245	2249	2279	2282	2320*	2327	2351	2354	2359	2389*	2391	2421
		2430	2486*	2487	2493	2518*	2519	2526	2556*	2557	2613*	2614	2673*	2674
		2682	2715*	2716*	5937	5939	5941	5943	5946	5949	5952	5955	5958	5962
		5973	5977	5981	5985	5989	5993							
\$REG5	001202	280#	2021	2058	2098	2135	2173	2212	2251	2284	2361	2393	2432	2522*
		2523	2531	2616*	2617	5937	5939	5941	5943	5946	5949	5952	5955	5958
		5962	5989	5993										
\$REG6	001204	281#	2023	2060	2100	2137	2175	2214	2253	2286	2363	2395	2434	5937
		5939	5941	5943	5946	5949	5952	5955	5958	5962				
\$REG7	001206	282#	2025	2062	2102	2139	2177	2216	2255	2288	2365	2397	2436	5937
		5939	5941	5943	5946	5949	5952	5955	5958	5962				
\$R2A =	***** U	5352												
\$SAVRE =	***** U	5352												
\$SAVR6	026022	5365*	5373	5374*	5375*	5395*								
\$SCOPE	023664	517	4923#											
\$SETUP =	000037	502#	516	517	519	521	523	525	526	527	529	4924	4988	5022
		5029												
\$STUP =	177777	502#												
\$SVLAD	024070	4934	4963#											
\$SVPC =	000204	203#	208											
\$SMR =	167400	1#	12	315	316	317	526	527	529	530	639	661	679	695
		711	727	744	766	794	827	854	881	908	931	960	992	1013
		1034	1066	1096	1114	1132	1150	1168	1186	1204	1216	1228	1240	1252
		1264	1277	1289	1301	1313	1325	1337	1350	1362	1374	1386	1398	1410
		1423	1435	1447	1459	1471	1484	1496	1508	1520	1532	1546	1565	1584
		1603	1622	1641	1661	1673	1685	1697	1709	1721	1734	1746	1758	1770
		1782	1794	1807	1819	1831	1843	1855	1867	1880	1892	1904	1916	1928
		1941	1953	1965	1977	1989	2008	2040	2077	2117	2154	2192	2231	2270

	2303	2342	2380	2412	2451	2482	2514	2552	2579	2609	2636	2666	2703
	2734	2758	2788	2819	2886	2903	2920	2937	2954	2971	2988	3005	3022
	3039	3056	3074	3093	3112	3131	3150	3169	3188	3207	3226	3245	3264
	3284	3301	3318	3335	3352	3369	3386	3403	3420	3437	3454	3472	3491
	3510	3529	3548	3567	3586	3605	3624	3643	3662	3682	3699	3716	3733
	3750	3767	3785	3804	3823	3842	3861	3880	3900	3917	3934	3951	3968
	3985	4002	4019	4036	4053	4070	4088	4107	4126	4145	4164	4183	4202
	4221	4240	4259	4278	4295	4314	4331	4349	4368	4387	4406	4425	4444
	4463	4482	4501	4520	4539	4558	4577	4596	4615	4634	4653	4672	4691
	4710	4729	4748	4767	4786	4805	4824	4843	4862	4881	4900	4919	4938
	4957	4976	4995	5014	5033	5052	5071	5090	5109	5128	5147	5166	5185
	5204	5223	5242	5261	5280	5299	5318	5337	5356	5375	5394	5413	5432
	5451	5470	5489	5508	5527	5546	5565	5584	5603	5622	5641	5660	5679
	5698	5717	5736	5755	5774	5793	5812	5831	5850	5869	5888	5907	5926
	5945	5964	5983	6002	6021	6040	6059	6078	6097	6116	6135	6154	6173
	6192	6211	6230	6249	6268	6287	6306	6325	6344	6363	6382	6401	6420
	6439	6458	6477	6496	6515	6534	6553	6572	6591	6610	6629	6648	6667
	6686	6705	6724	6743	6762	6781	6800	6819	6838	6857	6876	6895	6914
	6933	6952	6971	6990	7009	7028	7047	7066	7085	7104	7123	7142	7161
	7180	7199	7218	7237	7256	7275	7294	7313	7332	7351	7370	7389	7408
	7427	7446	7465	7484	7503	7522	7541	7560	7579	7598	7617	7636	7655
	7674	7693	7712	7731	7750	7769	7788	7807	7826	7845	7864	7883	7902
	7921	7940	7959	7978	7997	8016	8035	8054	8073	8092	8111	8130	8149
	8168	8187	8206	8225	8244	8263	8282	8301	8320	8339	8358	8377	8396
	8415	8434	8453	8472	8491	8510	8529	8548	8567	8586	8605	8624	8643
	8662	8681	8700	8719	8738	8757	8776	8795	8814	8833	8852	8871	8890
	8909	8928	8947	8966	8985	9004	9023	9042	9061	9080	9099	9118	9137
	9156	9175	9194	9213	9232	9251	9270	9289	9308	9327	9346	9365	9384
	9403	9422	9441	9460	9479	9498	9517	9536	9555	9574	9593	9612	9631
	9650	9669	9688	9707	9726	9745	9764	9783	9802	9821	9840	9859	9878
	9897	9916	9935	9954	9973	9992	10011	10030	10049	10068	10087	10106	10125
	10144	10163	10182	10201	10220	10239	10258	10277	10296	10315	10334	10353	10372
	10391	10410	10429	10448	10467	10486	10505	10524	10543	10562	10581	10600	10619
	10638	10657	10676	10695	10714	10733	10752	10771	10790	10809	10828	10847	10866
	10885	10904	10923	10942	10961	10980	10999	11018	11037	11056	11075	11094	11113
	11132	11151	11170	11189	11208	11227	11246	11265	11284	11303	11322	11341	11360
	11379	11398	11417	11436	11455	11474	11493	11512	11531	11550	11569	11588	11607
	11626	11645	11664	11683	11702	11721	11740	11759	11778	11797	11816	11835	11854
	11873	11892	11911	11930	11949	11968	11987	12006	12025	12044	12063	12082	12101
	12120	12139	12158	12177	12196	12215	12234	12253	12272	12291	12310	12329	12348
	12367	12386	12405	12424	12443	12462	12481	12500	12519	12538	12557	12576	12595
	12614	12633	12652	12671	12690	12709	12728	12747	12766	12785	12804	12823	12842
	12861	12880	12899	12918	12937	12956	12975	12994	13013	13032	13051	13070	13089
	13108	13127	13146	13165	13184	13203	13222	13241	13260	13279	13298	13317	13336
	13355	13374	13393	13412	13431	13450	13469	13488	13507	13526	13545	13564	13583
	13602	13621	13640	13659	13678	13697	13716	13735	13754	13773	13792	13811	13830
	13849	13868	13887	13906	13925	13944	13963	13982	14001	14020	14039	14058	14077
	14096	14115	14134	14153	14172	14191	14210	14229	14248	14267	14286	14305	14324
	14343	14362	14381	14400	14419	14438	14457	14476	14495	14514	14533	14552	14571
	14590	14609	14628	14647	14666	14685	14704	14723	14742	14761	14780	14799	14818
	14837	14856	14875	14894	14913	14932	14951	14970	14989	15008	15027	15046	15065
	15084	15103	15122	15141	15160	15179	15198	15217	15236	15255	15274	15293	15312
	15331	15350	15369	15388	15407	15426	15445	15464	15483	15502	15521	15540	15559
	15578	15597	15616	15635	15654	15673	15692	15711	15730	15749	15768	15787	15806
	15825	15844	15863	15882	15901	15920	15939	15958	15977	15996	16015	16034	16053
	16072	16091	16110	16129	16148	16167	16186	16205	16224	16243	16262	16281	16300
	16319	16338	16357	16376	16395	16414	16433	16452	16471	16490	16509	16528	16547
	16566	16585	16604	16623	16642	16661	16680	16699	16718	16737	16756	16775	16794
	16813	16832	16851	16870	16889	16908	16927	16946	16965	16984	17003	17022	17041
	17060	17079	17098	17117	17136	17155	17174	17193	17212	17231	17250	17269	17288
	17307	17326	17345	17364	17383	17402	17421	17440	17459	17478	17497	17516	17535
	17554	17573	17592	17611	17630	17649	17668	17687	17706	17725	17744	17763	17782
	17801	17820	17839	17858	17877	17896	17915	17934	17953	17972	17991	18010	18029
	18048	18067	18086	18105	18124	18143	18162	18181	18200	18219	18238	18257	18276
	18295	18314	18333	18352	18371	18390	18409	18428	18447	18466	18485	18504	18523
	18542	18561	18580	18599	18618	18637	18656	18675	18694	18713	18732	18751	18770
	18789	18808	18827	18846	18865	18884	18903	18922	18941	18960	18979	18998	19017
	19036	19055	19074	19093	19112	19131	19150	19169	19188	19207	19226	19245	19264
	19283	19302	19321	19340	19359	19378	19397	19416	19435	19454	19473	19492	19511
	19530	19549	19568	19587	19606	19625	19644	19663	19682	19701	19720	19739	19758
	19777	19796	19815	19834	19853	19872	19891	19910	19929	19948	19967	19986	20005
	20024	20043	20062	20081	20100	20119	20138	20157	20176	20195	20214	20233	20252
	20271	20290	20309	20328	20347	20366	20385	20404	20423	20442	20461	20480	20499
	20518	20537	20556	20575	20594	20613	20632	20651	20670	20689	20708	20727	20746
	20765	20784	20803	20822	20841	20860	20879	20898	20917	20936	20955	20974	20993
	21012	21031	21050	21069	21088	21107	21126	21145	21164	21183	21202	21221	21240
	21259	21278	21297	21316	21335	21354	21373	21392	21411	21430	21449	21468	21487
	21506	21525	21544	21563	21582	21601	21620	21639	21658	21677	21696	21715	21734
	21753	21772	21791	21810	21829	21848	21867	21886	21905	21924	21943	21962	21981
	21999	22018	22037	22056	22075	22094	22113	22132	22151	22170	22189	22208	22227
	22246	22265	22284	22303	22322	22341	22360	22379	22398	22417	22436	22455	22474
	22493	22512	22531	22550	22569	22588	22607	22626	22645	22664	22683	22702	22721
	22740	22759	22778	22797	22816	22835	22854	22873	22892	22911	22930	22949	22968
	22987	23006	23025	23044	23063	23082	23101	23120	23139	23158	23177	23196	23215
	23234	23253	23272	23291	23310	23329	23348	23367	23386	23405	23424	23443	23462
	23481	23500	23519	23538	23557	23576	23595	23614	23633	23652	23671	23690	23709
	23728	23747	23766	23785	23804	23823	23842	23861	23880	23899	23918	23937	23956
	23975	23994	24013	24032	24051	24070	24089	24108	24127	24146	24165	24184	24203
	24222	24241	24260	24279	24298	24317	24336	24355	24374	24393	24412	24431	24450
	24469	24488	24507	24526	24545	24564	24583	24602	24621	24640	24659	24678	24697
	24716	24735	24754	24773	24792	24811	24830	24849	24868	24887	24906	2492	



1798	1803	1807#	1811	1815	1819#	1823	1827	1831#	1835	1839	1843#	1847
1851	1855#	1859	1863	1867#	1871	1876	1880#	1884	1888	1892#	1896	1900
1904#	1908	1912	1916#	1920	1924	1928#	1932	1937	1941#	1945	1949	1953#
1957	1961	1965#	1969	1973	1977#	1981	1985	1989#	1993	2004	2008#	2030
2036	2040#	2067	2073	2077#	2107	2113	2117#	2144	2150	2154#	2182	2188
2192#	2221	2227	2231#	2260	2266	2270#	2293	2299	2303#	2332	2338	2342#
2370	2376#	2380	2402	2408	2412#	2441	2447	2451#	2473	2478	2482#	2505
2510	2514#	2543	2548	2552#	2570	2575	2579#	2600	2605	2609#	2627	2632
2636#	2657	2662	2666#	2694	2699	2703#	2725	2730	2734#	2749	2754	2758#
2779	2784	2788#	2805	2815	2819#	2881	2886	2889#	2898	2903#	2906	2915
2920#	2923	2932	2937#	2940	2949	2954#	2957	2966	2971#	2974	2983	2988#
2991	3000	3005#	3008	3017	3022#	3025	3034	3039#	3042	3051	3056#	3059
3069	3074#	3077	3088	3093#	3096	3107	3112#	3115	3126	3131#	3134	3145
3150#	3153	3164	3169#	3172	3183	3188#	3191	3202	3207#	3210	3221	3226#
3229	3240	3245#	3248	3259	3264#	3267	3279	3284#	3287	3296	3301#	3304
3313	3318#	3321	3330	3335#	3338	3347	3352#	3355	3364	3369#	3372	3381
3386#	3389	3398	3403#	3406	3415	3420#	3423	3432	3437#	3440	3449	3454#
3457	3467	3472#	3475	3486	3491#	3494	3505	3510#	3513	3524	3529#	3532
3543	3548#	3551	3562	3567#	3570	3581	3586#	3589	3600	3605#	3608	3619
3624#	3627	3638	3643#	3646	3657	3662#	3665	3677	3682#	3685	3694	3699#
3702	3711	3716#	3719	3728	3733#	3736	3745	3750#	3753	3762	3767#	3770
3780	3785#	3788	3799	3804#	3807	3818	3823#	3826	3837	3842#	3845	3856
3861#	3864	3875	3880#	3883	3895	3900#	3903	3912	3917#	3920	3929	3934#
3937	3946	3951#	3954	3963	3968#	3971	3980	3985#	3988	3997	4002#	4005
4014	4019#	4022	4031	4036#	4039	4048	4053#	4056	4065	4070#	4073	4083
4088#	4091	4102	4107#	4110	4121	4126#	4129	4140	4145#	4148	4159	4164#
4167	4178	4183#	4186	4197	4202#	4205	4216	4221#	4224	4235	4240#	4243
4254	4259#	4262	4273	4278#	4281	4293						
268#	5170#	5181										
272#	5119	5181										
267#	5168	5181										
521	5323#											
5334#	5345											
5338#	5347#	5348#	5349#	5350#								
5328	5345#											
228#												
243#	4300#	4345#	4941	4963#	4964	4969	4973	4999	5034			
5350												
5350												
5119#	5212	5338	5346									
5149	5156	5163	5168#	5169								
5009	5048#											
5174	5176	5179#										
5268#	5347											
5267	5270#	5349										
5263#	5348											
332#												
230#												
339#												
4928#												
5264#	5268#	5278	5313#									
4925	5009											
189#	193#	203	204#	206#	208#	209#	215	216#	218#	220#	239#	321
503#	514	529	530	4378	4453	4541	4584	4632	4675	4723	4766	4815
4858	4972	4973	5034	5101#	5181	5233#	5368	5394				
5185	5188											

STPB 001160  
 STPFLG 001165  
 STPS 001156  
 STRAP 025576  
 STRAP2 025620  
 STRP = 000005  
 STRPAD 025632  
 STSTM 001004  
 STSTM 001102  
 STYPB# \*\*\*\*\* U  
 STYPDS# \*\*\*\*\* U  
 STYPE 024620  
 STYPEC 025032  
 STYPER 024404  
 STYPEX 025100  
 STYPOC 025374  
 STYPON 025410  
 STYPOS 025350  
 SUNIT 001336  
 SUNITM 001010  
 SUSWR 001350  
 SXTSTR 023674  
 SOFILL 025573  
 \$40CAT= \*\*\*\*\* U  
 = 034074  
 \$ASTA= \*\*\*\*\* U

N13

FPU BASIC INSTR TESTS MACY11 27(1006) 25-APR-77 09:12 PAGE 151  
DQFPAB.P11 19-APR-77 13:36 CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0154

.SX = 001000 215# 220

CMPLT	18	4402	4477	4567	4612	4658	4703	4749	4794	4841	4886				
COMMEN	1418														
COMM00	18														
COMM01	18														
COMM02	18														
COMM03	18														
COMM04	18														
COMM05	18														
COMM06	18														
COMM07	18														
COMM1	18	2917	2934	2985	3002	3071	3128	3147	3204	3223	3281	3315	3349	3383	3488
	3526	3564	3602	3621	3640	3696	3730	3764	3782	3839	3858	3897	3931	3965	3999
	4033	4104	4142	4180	4218	4256									
COMM10	18														
COMM11	18														
COMM12	18														
COMM13	18														
COMM14	18														
COMM15	18														
COMM16	18														
COMM17	18														
COMM2	18	3053	3451	4067											
COMM20	18														
COMM21	18														
COMM22	18														
COMM23	18														
COMM24	18														
COMM25	18														
COMM26	18														
COMM27	18														
COMM3	18	2883	2900	2951	2968	3019	3036	3090	3109	3166	3185	3242	3298	3332	3366
	3400	3417	3434	3469	3507	3545	3583	3679	3713	3747	3801	3820	3877	3914	3948
	3982	4016	4050	4085	4123	4161	4199	4237							
COMM30	18														
COMM31	18														
COMM32	18														
COMM33	18														
COMM34	18														
COMM35	18														
COMM36	18														
COMM37	18														
COMM4	18	3261	3659	4275											
COMM40	18														
COMM41	18														
COMM42	18														
COMM43	18														
COMM44	18														
COMM45	18														
COMM46	18														
COMM47	18														
ENDCOM	1418														
ERRCMP	18	752	775	780	803	808	813	835	840	862	867	889	894	917	946
	973	978	999	1020	1042	1047	2051	2091	2128	2166	2205	2322	2327	2354	2425
	2461	2493	2526	2531	2588	2645	2677	2682	2774						
ERRLUR	18	5033													
ERROR	358	185	649	671	684	700	716	732	754	777	792	805	810	815	837