KD11-K

PDP11/6X FP11E FP BASIC

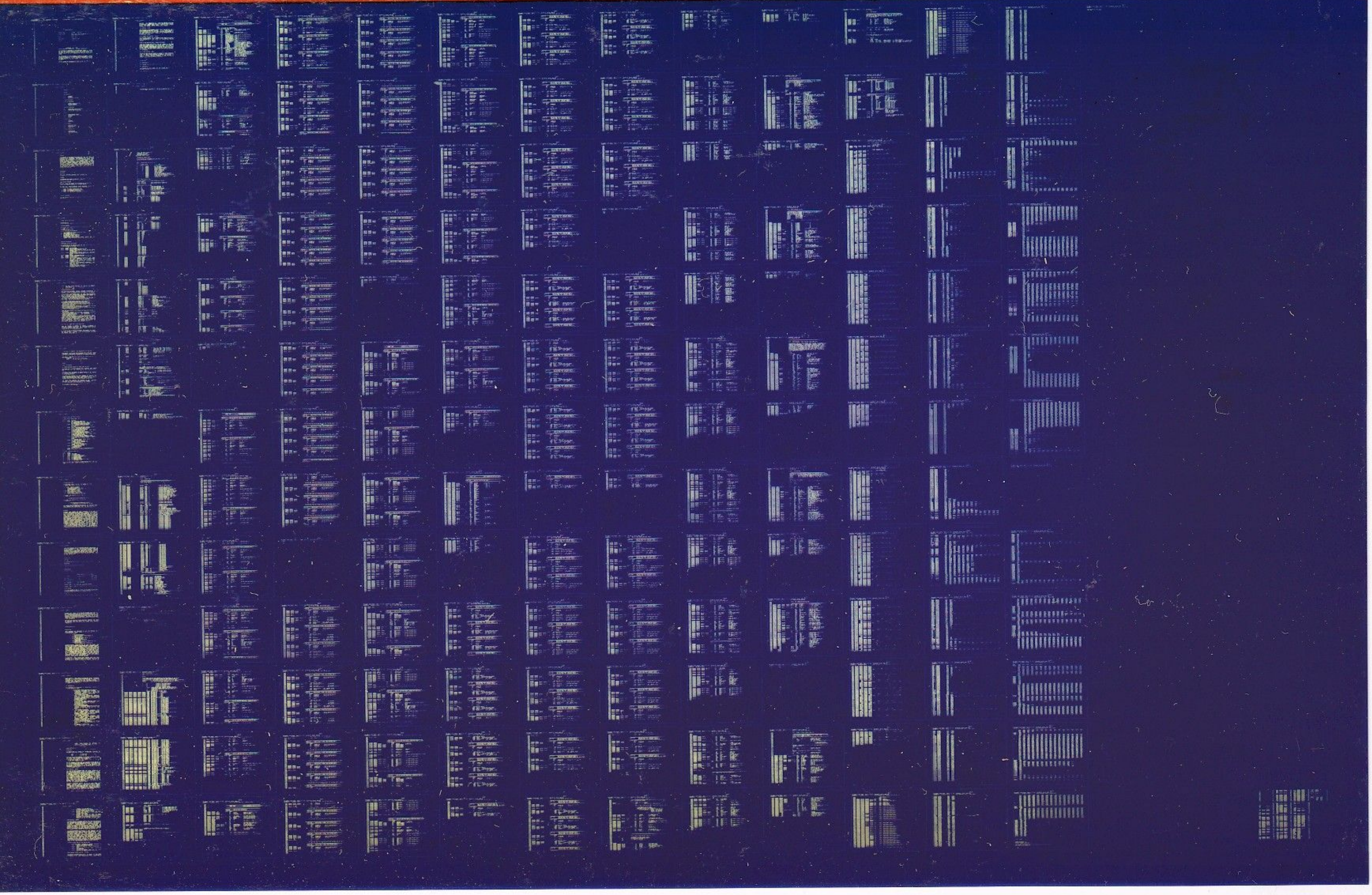
MD-11-DQFPA-A

EP-DQFPA-A-DL-A

FICHE 1 OF 1

APR 1977

MADE IN US



```
                000000          .REPT   0
```

IDENTIFICATION

PRODUCT CODE: MAINDEC-11-DQFPA-A-D

PRODUCT NAME: PDP-11/6X - FP11-E FLOATING POINT UNIT
BASIC INSTRUCTION TESTS

DATE : MARCH 1977

MAINTAINER: DIAGNOSTIC GROUP

AUTHOR: DONALD NORTH

CONTENTS
----

1. ABSTRACT

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

2. REQUIREMENTS

2.1 EQUIPMENT

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

2.2 STORAGE

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

2.3 PRELIMINARY PROGRAMS

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

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

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

3. LOADING PROCEDURE

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

4. STARTING PROCEDURE

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 "$LPTST" IF SET, THEN THE TEST SPECIFIED BY THE TEST NUMBER CONTAINED IN THE MEMORY WORD "$LPTST" (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) SET=TEST ONLY UNIT SPECIFIED IN SW00
SW00	000001	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 "$LPTST" MATCHES THE NUMBER OF THE TEST CURRENTLY EXECUTING. AT THIS POINT, THE TEST IS LOOPED ON INDEFINATELY, UNTIL EITHER SW<8>=0 OR "$LPTST" IS CHANGED. NOTE THAT IF "$LPTST" DOES NOT MATCH THE TEST NUMBER OF ANY TEST, THE CONTENTS OF "$LPTST" 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 HFP FP11-E OPTION ONLY
SW00=1 --> TEST WFP PDP-11/6X MICROCODE ONLY

6. ERRORS

6.1 FORMAT OF MESSAGES

6.1.1 ALL ERROR MESSAGES CONSIST OF THREE LINES OF DATA:

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

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

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

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

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

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	
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.

```
                        FLOATING POINT DATA:

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

                        FIRST WORD:  (BOTH F, D MODES)
                                B15=SIGN OF NUMBER (1/-, 0/+)
                                B14:07=EXPONENT, 8.BITS, FROM -128./+127.
                                B06:00=FRACTION, 7.BITS
                        SECOND WORD:  (BOTH F, D MODES)
                                B15:00=FRACTION, 16.BITS
                        THIRD, FOURTH WORDS:  (ONLY D MODE)
                                B15:00, B15:00=FRACTION, 32. BITS

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

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

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

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

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


                6.2     RECOVERY

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

                6.3     CAUSES
```

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

7. RESTRICTIONS

7.1 STARTING

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

7.2 OPERATIONAL

THERE ARE NO OPERATIONAL RESTRICTIONS.

8. MISCELLANEOUS

8.1 EXECUTION TIME

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

SEC = SECONDS / MIN = MINUTES

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

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

8.2 STACK POINTER

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

8.3 POWER FAIL

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

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 SECIONS 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 AC0-5.

(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 (-D 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))

$TSTNM - A COUNTER INDICATING THE NUMBER (1-377(8)) OF THE TEST CURRENTLY BEING EXECUTED

$LPADR - CONTAINS THE ADDRESS TO WHICH THE SCOPE ROUTINE 10240 WILL LOOP, IF THE CURRENT TEST IS BEING LOOPED UPON

$LPERR - 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 $LPADR, ABOVE.

9.3.3 ERROR ROUTINE - $ERROR

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 SIGNALED. 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 ($ERRTB) 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

$ERTTL - COMULATIVE NUMBER OF ERRORS ENCOUNTERED TO DATE

$ERRPC - CONTAINS THE PC OF THE "ERROR" INSTRUCTION JUST EXECUTED

$LPERR - CONTAINS THE ADDRESS WHICH WILL BE LOOPED UPON FOR THE ERROR LOOPING FACILITY

9.3.4 ERROR MESSAGE TYPEOUT ROUTINE - $TYPERR

THIS ROUTINE ($TYPERR 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 SERRTB (ERROR TABLE), AND TYPE OUT THE ERROR MESSAGE, DATA HEADER, AND DATA VALUES ON THE CONSOLE.

9.3.5 TYPE ROUTINE - STYPE

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

9.3.6 OCTAL NUMBER TYPE ROUTINE - STYPOC

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

9.3.7 POWER UP AND DOWN ROUTINES - SPWRUP AND SPWRDN

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

9.3.8 END OF PASS ROUTINE - SEOP

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

10. ACT/APT/XXDP

10.1 ACT COMPATIBILITY

THIS PROGRAM WILL RUN UNDER THE ACT SYSTEM.

10.2 APT COMPATIBILITY

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

10.3 XXDP COMPATIBILITY

```
                                MAINDEC-11-DQFPA-A                                PAGE 15


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



                        .ENDR
```

```
                                .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
                                .EQUIV  PS,PSW
177774                          STKLMT= 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
```

```
000006                  R6=     %6              ;;GENERAL REGISTER
000007                  R7=     %7              ;;GENERAL REGISTER
000006                  SP=     %6              ;;STACK POINTER
000007                  PC=     %7              ;;PROGRAM COUNTER

                        ;*PRIORITY LEVEL DEFINITIONS
000000                  PR0=    0               ;;PRIORITY LEVEL 0
000040                  PR1=    40              ;;PRIORITY LEVEL 1
000100                  PR2=    100             ;;PRIORITY LEVEL 2
000140                  PR3=    140             ;;PRIORITY LEVEL 3
000200                  PR4=    200             ;;PRIORITY LEVEL 4
000240                  PR5=    240             ;;PRIORITY LEVEL 5
000300                  PR6=    300             ;;PRIORITY LEVEL 6
000340                  PR7=    340             ;;PRIORITY LEVEL 7

                        ;*"SWITCH REGISTER" SWITCH DEFINITIONS
100000                  SW15=   !00000
040000                  SW14=   40000
020000                  SW13=   20000
010000                  SW12=   10000
004000                  SW11=   4000
002000                  SW10=   2000
001000                  SW09=   1000
000400                  SW08=   400
000200                  SW07=   200
000100                  SW06=   100
000040                  SW05=   40
000020                  SW04=   20
000010                  SW03=   10
000004                  SW02=   4
000002                  SW01=   2
000001                  SW00=   1
                        .EQUIV  SW09,SW9
                        .EQUIV  SW08,SW8
                        .EQUIV  SW07,SW7
                        .EQUIV  SW06,SW6
                        .EQUIV  SW05,SW5
                        .EQUIV  SW04,SW4
                        .EQUIV  SW03,SW3
                        .EQUIV  SW02,SW2
                        .EQUIV  SW01,SW1
                        .EQUIV  SW00,SW0

                        ;*DATA BIT DEFINITIONS (BIT00 TO BIT15)
100000                  BIT15=  100000
040000                  BIT14=  40000
020000                  BIT13=  20000
010000                  BIT12=  10000
004000                  BIT11=  4000
002000                  BIT10=  2000
001000                  BIT09=  1000
000400                  BIT08=  400
000200                  BIT07=  200
000100                  BIT06=  100
000040                  BIT05=  40
000020                  BIT04=  20
```

```
000010                  BIT03=  10
000004                  BIT02=  4
000002                  BIT01=  2
000001                  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
000004                  ERRVEC= 4               ;;TIME OUT AND OTHER ERRORS
000010                  RESVEC= 10              ;;RESERVED AND ILLEGAL INSTRUCTIONS
000014                  TBITVEC=14              ;;"T" BIT
000014                  TRTVEC= 14              ;;TRACE TRAP
000014                  BPTVEC= 14              ;;BREAKPOINT TRAP (BPT)
000020                  IOTVEC= 20              ;;INPUT/OUTPUT TRAP (IOT) **SCOPE**
000024                  PWRVEC= 24              ;;POWER FAIL
000030                  EMTVEC= 30              ;;EMULATOR TRAP (EMT) **ERROR**
000034                  TRAPVEC=34              ;;"TRAP" TRAP
000060                  TKVEC=  60              ;;TTY KEYBOARD VECTOR
000064                  TPVEC=  64              ;;TTY PRINTER VECTOR
000240                  PIRQVEC=240             ;;PROGRAM INTERRUPT REQUEST VECTOR

                        ;*MED INSTR CODES
076600                  MED=    076600          ;OPCODE

000022                  RWHAMI= 022             ;READ WHAMI

000144                  RFLAG=  144             ;READ FLAGS
000344                  WFLAG=  344             ;WRITE FLAGS

                        ;*FLOATING POINT INTERRUPT VECTOR
000244                  FPPVEC= 244

                        ;*FLOATING POINT REGISTER DEFINITIONS
000000                  AC0=    %0
000001                  AC1=    %1
000002                  AC2=    %2
000003                  AC3=    %3
000004                  AC4=    %4
000005                  AC5=    %5

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

```
                177777          LGN=    177777          ; 1111...11  LGST - NUM (1ST WD FLT )
                000200          SMP=    000200          ; +1*2**-128, SMLT + NUM (1ST WD FLT)
                100200          SMN=    100200          ; -1*2**-128, SMLT - NUM (1ST WD FLT)
                000177          ZX1MP=  000177          ; ZERO EXP, ALL 1-S MANT (1ST WD FLT)
                100177          ZX1MN=  100177          ; ZERO EXP, ALL 1-S MANT (1ST WD FLT)
                040200          F1P=    040200          ; +1.0E+0, 1ST WD FLT
                140200          F1N=    140200          ; -1.0E+0, 1ST WD FLT

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

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

                                ;MISC
                104117          ERROR117=       ERROR!117       ; FOR USE AS A LITERAL

                                .SBTTL  TRAP CATCHER

                000000                  .=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
                000174                  .=174
000174  000000  DISPREG: .WORD  0               ;;SOFTWARE DISPLAY REGISTER
000176  000000  SWREG:   .WORD  0               ;;SOFTWARE SWITCH REGISTER
                .SBTTL  STARTING ADDRESS(ES)
000200  000137  002506          JMP     @#START ;;JUMP TO STARTING ADDRESS OF PROGRAM

                 SBTTL  ACT11 HOOKS

                ;;*****************************************************************
                ;HOOKS REQUIRED BY ACT11
                000204          $SVPC=.                 ;SAVE PC
                000046          .=46
000046  020510          $ENDAD                  ;;1)SET LOC.46 TO ADDRESS OF $ENDAD IN .$EOP
                000052          .=52
000052  000000          .WORD   0               ;;2)SET LOC.52 TO ZERO
                000204          .=$SVPC                 ;; RESTORE PC
                001000          .=1000
                .SBTTL  APT PARAMETER BLOCK

                ;;*****************************************************************
                ;SET LOCATIONS 24 AND 44 AS REQUIRED FOR APT
                ;;*****************************************************************
                001000          .$X=.   ;;SAVE CURRENT LOCATION
                000024          .=24    ;;SET POWER FAIL TO POINT TO START OF PROGRAM
000024  000200          200     ;;FOR APT START UP
                000044          .=44    ;;POINT TO APT INDIRECT ADDRESS PNTR.
000044  001000          $APTHDR ;;POINT TO APT HEADER BLOCK
                001000          .=.$X   ;;RESET LOCATION COUNTER
                ;;*****************************************************************
                ;SETUP APT PARAMETER BLOCK AS DEFINED IN THE APT-PDP11 DIAGNOSTIC
                ;INTERFACE SPEC.
```

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

```
                                .SBTTL  COMMON TAGS

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

                        001100          .=1100
001100                  $CMTAG:                         ;;START OF COMMON TAGS
001100  000000                  .WORD   0
001102  000000          $TSTNM: .WORD   0               ;;CONTAINS THE TEST NUMBER
001104  000000          $ERFLG: .WORD   0               ;;CONTAINS ERROR FLAG
001106  000000          $ICNT:  .WORD   0               ;;CONTAINS SUBTEST ITERATION COUNT
001110  000000          $LPADR: .WORD   0               ;;CONTAINS SCOPE LOOP ADDRESS
001112  000000          $LPTST: .WORD   0               ;;CONTAINS TEST NUMBER TO LOOP UPON
001114  000000          $LPERR: .WORD   0               ;;CONTAINS SCOPE RETURN FOR ERRORS
001116  000000          $ERTTL: .WORD   0               ;;CONTAINS TOTAL ERRORS DETECTED
001120  000000          $ITEMB: .WORD   0               ;;CONTAINS ITEM CONTROL BYTE
001122  000001          $ERMAX: .WORD   1               ;;CONTAINS MAX. ERRORS PER TEST
001124  000000          $ERRPC: .WORD   0               ;;CONTAINS PC OF LAST ERROR INSTRUCTION
001126  000000          $GDADR: .WORD   0               ;;CONTAINS ADDRESS OF 'GOOD' DATA
001130  000000          $BDADR: .WORD   0               ;;CONTAINS ADDRESS OF 'BAD' DATA
001132  000000          $GDDAT: .WORD   0               ;;CONTAINS 'GOOD' DATA
001134  000000          $BDDAT: .WORD   0               ;;CONTAINS 'BAD' DATA
001136  000000                  .WORD   0               ;;RESERVED--NOT TO BE USED
001140  000000                  .WORD   0
001142  000             $AUTOB: .BYTE   0               ;;AUTOMATIC MODE INDICATOR
001143  000             $INTAG: .BYTE   0               ;;INTERRUPT MODE INDICATOR
001144  000000                  .WORD   0
001146  177570          SWR:    .WORD   DSWR            ;;ADDRESS OF SWITCH REGISTER
001150  177570          DISPLAY: .WORD  DDISP           ;;ADDRESS OF DISPLAY REGISTER
001152  177560          $TKS:   177560                  ;;TTY KBD STATUS
001154  177562          $TKB:   177562                  ;;TTY KBD BUFFER
001156  177564          $TPS:   177564                  ;;TTY PRINTER STATUS REG. ADDRESS
001160  177566          $TPB:   177566                  ;;TTY PRINTER BUFFER REG. ADDRESS
001162  000             $NULL:  .BYTE   0               ;;CONTAINS NULL CHARACTER FOR FILLS
001163  002             $FILLS: .BYTE   2               ;;CONTAINS # OF FILLER CHARACTERS REQUIRED
001164  012             $FILLC: .BYTE   12              ;;INSERT FILL CHARS. AFTER A "LINE FEED"
001165  000             $TPFLG: .BYTE   0               ;;"TERMINAL AVAILABLE" FLAG (BIT<07>=0=YES)
001166  000000          $REGAD: .WORD   0               ;;CONTAINS THE ADDRESS FROM
                                                        ;;WHICH  ($REG0) WAS OBTAINED
001170  000000          $REG0:  .WORD   0               ;;CONTAINS (($REGAD)+0)
001172  000000          $REG1:  .WORD   0               ;;CONTAINS (($REGAD)+2)
001174  000000          $REG2:  .WORD   0               ;;CONTAINS (($REGAD)+4)
001176  000000          $REG3:  .WORD   0               ;;CONTAINS (($REGAD)+6)
001200  000000          $REG4:  .WORD   0               ;;CONTAINS (($REGAD)+10)
001202  000000          $REG5:  .WORD   0               ;;CONTAINS (($REGAD)+12)
001204  000000          $REG6:  .WORD   0               ;;CONTAINS (($REGAD)+14)
001206  000000          $REG7:  .WORD   0               ;;CONTAINS (($REGAD)+16)
001210  000000          $REG10: .WORD   0               ;;CONTAINS (($REGAD)+20)
001212  000000          $REG11: .WORD   0               ;;CONTAINS (($REGAD)+22)
001214  000000          $REG12: .WORD   0               ;;CONTAINS (($REGAD)+24)
001216  000000          $REG13: .WORD   0               ;;CONTAINS (($REGAD)+26)
001220  000000          $REG14: .WORD   0               ;;CONTAINS (($REGAD)+30)
001222  000000          $REG15: .WORD   0               ;;CONTAINS (($REGAD)+32)
001224  000000          $REG16: .WORD   0               ;;CONTAINS (($REGAD)+34)
001226  000000          $REG17: .WORD   0               ;;CONTAINS (($REGAD)+36)
```

```
1052  001230  000000                  $TMP0:  .WORD   0               ;;USER DEFINED
1053  001232  000000                  $TMP1:  .WORD   0               ;;USER DEFINED
1054  001234  000000                  $TMP2:  .WORD   0               ;;USER DEFINED
1055  001236  000000                  $TMP3:  .WORD   0               ;;USER DEFINED
1056  001240  000000                  $TMP4:  .WORD   0               ;;USER DEFINED
1057  001242  000000                  $TMP5:  .WORD   0               ;;USER DEFINED
1058  001244  000000                  $TMP6:  .WORD   0               ;;USER DEFINED
1059  001246  000000                  $TMP7:  .WORD   0               ;;USER DEFINED
1060  001250  000000                  $TMP10: .WORD   0               ;;USER DEFINED
1061  001252  000000                  $TMP11: .WORD   0               ;;USER DEFINED
1062  001254  000000                  $TMP12: .WORD   0               ;;USER DEFINED
1063  001256  000000                  $TMP13: .WORD   0               ;;USER DEFINED
1064  001260  000000                  $TMP14: .WORD   0               ;;USER DEFINED
1065  001262  000000                  $TMP15: .WORD   0               ;;USER DEFINED
1066  001264  000000                  $TMP16: .WORD   0               ;;USER DEFINED
1067  001266  000000                  $TMP17: .WORD   0               ;;USER DEFINED
1068  001270  000000                  $TMP20: .WORD   0               ;;USER DEFINED
1069  001272  000000                  $TMP21: .WORD   0               ;;USER DEFINED
1070  001274  000000                  $TMP22: .WORD   0               ;;USER DEFINED
1071  001276  000000                  $TMP23: .WORD   0               ;;USER DEFINED
1072  001300  000000                  $TMP24: .WORD   0               ;;USER DEFINED
1073  001302  000000                  $TMP25: .WORD   0               ;;USER DEFINED
1074  001304  000000                  $TMP26: .WORD   0               ;;USER DEFINED
1075  001306  000000                  $TMP27: .WORD   0               ;;USER DEFINED
1076  001310  000000                  $TIMES: 0                       ;;MAX. NUMBER OF ITERATIONS
1077  001312  000000                  $ESCAPE:0                       ;;ESCAPE ON ERROR ADDRESS
1078  001314  177607  000377          $BELL:  .ASCIZ  <207><377><377> ;;CODE FOR BELL
1079  001320     077                  $QUES:  .ASCII  /?/             ;;QUESTION MARK
1080  001321     015                  $CRLF:  .ASCII  <15>            ;;CARRIAGE RETURN
1081  001322  000012                  $LF:    .ASCIZ  <12>            ;;LINE FEED
1082                                  ;;**************************************************************
1083                                  .SBTTL  APT MAILBOX-ETABLE
1084
1085                                  ;;**************************************************************
1086                                  .EVEN
1087  001324                          $MAIL:                          ;;APT MAILBOX
1088  001324  000000                  $MSGTY: .WORD   AMSGTY          ;;MESSAGE TYPE CODE
1089  001326  000000                  $FATAL: .WORD   AFATAL          ;;FATAL ERROR NUMBER
1090  001330  000000                  $TESTN: .WORD   ATESTN          ;;TEST NUMBER
1091  001332  000000                  $PASS:  .WORD   APASS           ;;PASS COUNT
1092  001334  000000                  $DEVCT: .WORD   ADEVCT          ;;DEVICE COUNT
1093  001336  000000                  $UNIT:  .WORD   AUNIT           ;;I/O UNIT NUMBER
1094  001340  000000                  $MSGAD: .WORD   AMSGAD          ;;MESSAGE ADDRESS
1095  001342  000000                  $MSGLG: .WORD   AMSGLG          ;;MESSAGE LENGTH
1096  001344                          $ETABLE:                        ;;APT ENVIRONMENT TABLE
1097  001344     000                  $ENV:   .BYTE   AENV            ;;ENVIRONMENT BYTE
1098  001345     000                  $ENVM:  .BYTE   AENVM           ;;ENVIRONMENT MODE BITS
1099  001346  000000                  $SWREG: .WORD   ASWREG          ;;APT SWITCH REGISTER
1100  001350  000000                  $USWR:  .WORD   AUSWR           ;;USER SWITCHES
1101  001352  000000                  $CPUOP: .WORD   ACPUOP          ;;CPU TYPE,OPTIONS
1102                                  ;*                              BITS 15-11=CPU TYPE
1103                                  ;*                                      11/04=01,11/05=02,11/20=03,11/40=04,11/45=05
1104                                  ;*                                      11/70=06,PDQ=07,Q=10
1105                                  ;*                              BIT 10=REAL TIME CLOCK
1106                                  ;*                              BIT  9=FLOATING POINT PROCESSOR
1107                                  ;*                              BIT  8=MEMORY MANAGEMENT
```

```
 1108  001354                          $ETEND:
 1109                                  .MEXIT
```

```
                                        .SBTTL  ERROR POINTER TABLE

001354                                  $ERRTB:

                                        ;*THIS TABLE CONTAINS THE INFORMATION FOR EACH ERROR THAT CAN OCCUR.
                                        ;*THE INFORMATION IS OBTAINED BY USING THE INDEX NUMBER FOUND IN
                                        ;*LOCATION $ITEMB. THIS NUMBER INDICATES WHICH ITEM IN THE TABLE IS PERTINENT.
                                        ;*NOTE1:        IF $ITEMB IS 0 THE ONLY PERTINENT DATA IS ($ERRPC).
                                        ;*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 ($ERRTB) HAS BEEN MODIFIED,
                                                ;*       ELIMINATING UNUSED VALUE FOR DATA FORMAT POINTER.
                                                ;*       ERROR TYPING ROUTINE HAS ALSO BEEN MODIFIED
                                                ;*       ACCORDINGLY.
                                                ;*****  SPECIFIC TEST VECTORS  *****
001354  026023  030215  032620  EMV001: .WORD   EMF,DHA,DTB      ; FPS RIPPLE 1
001362  026060  030215  032620  EMV002: .WORD   EMG,DHA,DTB      ; FPS RIPPLE 0
001370  026115  030233  032614  EMV003: .WORD   EMH,DHB,DTA      ; SETF
001376  026154  030233  032614  EMV004: .WORD   EMI,DHB,DTA      ; SETD
001404  026211  030233  032614  EMV005: .WORD   EMJ,DHB,DTA      ; SETI
001412  026250  030233  032614  EMV006: .WORD   EMK,DHB,DTA      ; SETL
                                                ;*****  SRC/DST ADDR MODES VECTORS  *****
001420  026305  030765  033104  EMV007: .WORD   EML,DHQ,DTT      ; M05-M77 RESULT
001426  026477  031054  033132  EMV010: .WORD   EMN,DHQ,DTV      ; M13-M35 DSTREG
001434  026305  031054  033132  EMV011: .WORD   EML,DHQ,DTV      ; M13-M35 RESULT
001442  026375  031016  033116  EMV012: .WORD   EMM,DHP,DTU      ; M21-M53 SRCREG
001450  026477  031016  033116  EMV013: .WORD   EMN,DHP,DTU      ; M21-M53 DSTREG
001456  026305  031016  033116  EMV014: .WORD   EML,DHP,DTU      ; M21-M53 RESULT, M63-M71 RESULT
001464  026375  030652  033046  EMV015: .WORD   EMM,DHL,DTQ      ; M30-M67 SRCREG
001472  026305  030652  033046  EMV016: .WORD   EML,DHL,DTQ      ; M30-M67 RESULT
001500  026375  030622  033034  EMV017: .WORD   EMM,DHK,DTP      ; M42-M00 SRCREG
001506  026305  030622  033034  EMV020: .WORD   EML,DHK,DTP      ; M42-M00 RESULT
001514  026375  030734  033072  EMV021: .WORD   EMM,DHN,DTS      ; M54-M37 SRCREG
001522  026305  030734  033072  EMV022: .WORD   EML,DHN,DTS      ; M54-M37 RESULT, M77-M24 RESULT
001530  026305  030703  033060  EMV023: .WORD   EML,DHM,DTR      ; M72-M27 RESULT
001536  026477  030577  033024  EMV024: .WORD   EMN,DHJ,DTO      ; M27-M43 DSTREG
001544  026305  030577  033024  EMV025: .WORD   EML,DHJ,DTO      ; M27-M43 RESULT
001552  026305  030554  033014  EMV026: .WORD   EML,DHI,DTN      ; M37-M62 RESULT, M67-M12 RESULT
001560  026477  030734  033072  EMV027: .WORD   EMN,DHN,DTS      ; M77-M24 DSTREG
                                                ;*****  VECTOR FOR CFCC TEST  *****
001566  026601  030215  032620  EMV030: .WORD   EMO,DHA,DTB      ; CFCC INSTR
                                                ;*****  FSRC/FDST D ADDR MODES VECTORS  *****
001574  026657  031215  033174  EMV031: .WORD   EMP,DHT,DTY      ; M15-M67 RESULT, M67-M25 RESULT
001602  026753  031151  033160  EMV032: .WORD   EMQ,DHS,DTX      ; M44-M37 SRCREG
001610  026657  031151  033160  EMV033: .WORD   EMP,DHS,DTX      ; M44-M37 RESULT
001616  027062  031666  033336  EMV034: .WORD   EMR,DHAA,DTAF    ; M75-M34 DSTREG
001624  027057  031666  033336  EMV035: .WORD   EMP,DHAA,DTAF    ; M75-M34 RESULT
001632  027062  031215  033174  EMV036: .WORD   EMR,DHT,DTY      ; M67-M25 DSTREG
001640  026753  031261  033210  EMV037: .WORD   EMQ,DHU,DTZ      ; M20-M13 SRCREG
001646  026657  031261  033210  EMV040: .WORD   EMP,DHU,DTZ      ; M20-M13 RESULT
001654  026753  031604  033316  EMV041: .WORD   EMQ,DHZ,DTAE     ; M51-M77 SRCREG
001662  026657  031604  033316  EMV042: .WORD   EMP,DHZ,DTAE     ; M51-M77 RESULT
```

```
1166  001670  026657  031403  033244  EMV043: .WORD   EMP,DHW,DTAB    ; M27-M70 RESULT
1167  001676  026657  031531  033300  EMV044: .WORD   EMP,DHY,DTAD    ; M77-M64 RESULT
1168  001704  026753  031456  033262  EMV045: .WORD   EMQ,DHX,DTAC    ; M32-M27 SRCREG
1169  001712  026657  031456  033262  EMV046: .WORD   EMP,DHX,DTAC    ; M32-M27 RESULT, M00-M52 RESULT
1170  001720  027062  031332  033226  EMV047: .WORD   EMR,DHV,DTAA    ; M63-MH1 DSTREG
1171  001726  026657  031332  033226  EMV050: .WORD   EMP,DHV,DTAA    ; M63-MH1 RESULT
1172  001734  026657  031112  033146  EMV051: .WORD   EMP,DHR,DTW     ; M37-M03 RESULT
1173  001742  027062  031456  033262  EMV052: .WORD   EMR,DHX,DTAC    ; M00-M52 DSTREG
1174                                  ;***** FSRC/FDST F ADDR MODES VECTORS  *****
1175  001750  027374  032063  033410  EMV053: .WORD   EMU,DHAO,DTAI    ; M12-MH5 DSTREG
1176  001756  027171  032063  033410  EMV054: .WORD   EMS,DHAO,DTAI    ; M12-MH5 RESULT
1177  001764  027265  032177  033442  EMV055: .WORD   EMT,DHAF,DTAK    ; M34-M60 SRCREG
1178  001772  027171  032177  033442  EMV056: .WORD   EMS,DHAF,DTAK    ; M34-M60 RESULT
1179  002000  027265  032470  033542  EMV057: .WORD   EMT,DHAJ,DTAO    ; M50-M32 SRCREG
1180  002006  027374  032470  033542  EMV060: .WORD   EMU,DHAJ,DTAO    ; M50-M32 DSTREG
1181  002014  027171  032470  033542  EMV061: .WORD   EMS,DHAJ,DTAO    ; M50-M32 RESULT
1182  002022  027171  032255  033462  EMV062: .WORD   EMS,DHAG,DTAL    ; M72-M11 RESULT
1183  002030  027374  032020  033374  EMV063: .WORD   EMU,DHAC,DTAH    ; M37-M23 DSTREG
1184  002036  027171  032020  033374  EMV064: .WORD   EMS,DHAC,DTAH    ; M37-M23 RESULT, M23-M37 RESULT
1185  002044  027171  032411  033522  EMV065: .WORD   EMS,DHAJ,DTAH    ; M77-M75 RESULT
1186  002052  027265  032020  033374  EMV066: .WORD   EMT,DHAC,DTAH    ; M23-M37 SRCREG
1187  002060  027265  032333  033502  EMV067: .WORD   EMT,DHAH,DTAH    ; MH5-MH4 SRCREG
1188  002066  027374  032333  033502  EMV070: .WORD   EMU,DHAH,DTAH    ; MH5-MH4 DSTREG
1189  002074  027171  032333  033502  EMV071: .WORD   EMS,DHAH,DTAH    ; MH5-MH4 RESULT
1190  002102  027171  032133  033226  EMV072: .WORD   EMS,DHAE,DTAJ    ; M27-M77 RESULT
1191  002110  027171  032554  033526  EMV073: .WORD   EMS,DHAK,DTC     ; M57-M67 RESULT, M02-M27 RESULT
1192  002116  027171  031755  033360  EMV074: .WORD   EMS,DHAB,DTAG    ; M61-M01 RESULT
1193                                  ;***** FP ACC ALL THERE ? VECTOR  *****
1194  002124  025752  030242  032626  EMV075: .WORD   EMF,DHC,DTC      ; FP AC ALL THERE  ?
1195                                  ;***** FPS ERROR VECTORS  *****
1196  002132  025543  030300  032640  EMV076: .WORD   EMA,DHD,DTD      ; FPS - ABS,NEG,CLR,TST F
1197  002140  025543  030300  032646  EMV077: .WORD   EMA,DHD,DTE      ; FPS - LDD/STD, ABS,NEG,CLR,TST D
1198  002146  025543  030300  032654  EMV100: .WORD   EMA,DHD,DTF      ; FPS - LDD/LDF/STD
1199                                  ;***** FEC/FEA ERROR VECTORS  *****
1200  002154  025567  030314  032662  EMV101: .WORD   EMB,DHE,DTG      ; FEC/FEA - ABS,NEG,CLR,TST F
1201  002162  025567  030314  032674  EMV102: .WORD   EMB,DHE,DTH      ; FEC/FEA - LDD/STD, ABS,....,CLR D
1202  002170  025567  030314  032706  EMV103: .WORD   EMB,DHE,DTI      ; FEC/FEA - LDD/LDF/STD
1203                                  ;***** RESULT VECTORS  *****
1204  002176  027503  030412  032732  EMV104: .WORD   EMV,DHG,DTK      ; RESULT - LDD/STD
1205  002204  027541  030412  032754  EMV105: .WORD   EMW,DHG,DTL      ; RESULT - LDD/LDF/STD
1206  002212  025617  030354  033020  EMV106: .WORD   EMC,DHF,DTJ      ; RESULT - ABSF
1207  002220  025617  030412  033032  EMV107: .WORD   EMC,DHG,DTK      ; RESULT - ABSD
1208  002226  027603  030354  033120  EMV110: .WORD   EMX,DHF,DTJ      ; RESULT - NEGF
1209  002234  027603  030412  033132  EMV111: .WORD   EMX,DHG,DTK      ; RESULT - NEGD
1210  002242  027737  030354  033120  EMV112: .WORD   EMZ,DHF,DTJ      ; RESULT - CLRF
1211  002250  027737  030412  033132  EMV113: .WORD   EMZ,DHG,DTK      ; RESULT - CLRD
1212  002256  027651  030354  033210  EMV114: .WORD   EMY,DHF,DTJ      ; RESULT - TSTF
1213  002264  027651  030412  033222  EMV115: .WORD   EMY,DHG,DTK      ; RESULT - TSTD
1214                                  ;***** ILLEGAL TRAP CATCHER VECTOR  *****
1215  002272  025665  030510  032776  EMV116: .WORD   EMD,DHH,DTM      ; ILLEGAL TRAP CATCHER
1216                                  ;***** PC MODE 2 WRONG INCRE VECTORS *****
1217  002300  030005  032612  033504  EMV117: .WORD   EMAA,DHAL,DTAP   ;+0
1218  002306  030042  032612  033514  EMV120: .WORD   EMAB,DHAL,DTAP   ;+4
1219  002314  030113  032612  033524  EMV121: .WORD   EMAC,DHAL,DTAP   ;+6
1220  002322  030157  032612  033534  EMV122: .WORD   EMAD,DHAL,DTAP   ;+10
```

```
                                        .SBTTL  PROGRAM DEFINED COMMON TAGS
                                        ;#VARIABLES
002330  000000                  FPS:    .WORD   0               ; FPS STORED HERE AFTER STFPS
002332  000000                  FEC:    .WORD   0               ; FEC STORED HERE AFTER STST
002334  000000                  FEA:    .WORD   0               ; FEA STORED HERE AFTER STST
002336  000000                  FPPOPC: .WORD   0               ; OLD PC SAVED HERE AFTER TRAP
002340  000000                  FPPOPS: .WORD   0               ; OLD PS SAVED HERE AFTER TRAP
002342  000000                  FPPOSP: .WORD   0               ; SP AFTER TRAP
002344  000000                  EXPFEA: .WORD   0               ; EXPECTED FEA

                                        ;#REGISTER CONTENTS, AT ERROR, STORED HERE
002346  000000                  EREG0:  .WORD   0
002350  000000                  EREG1:  .WORD   0
002352  000000                  EREG2:  .WORD   0
002354  000000                  EREG3:  .WORD   0
002356  000000                  EREG4:  .WORD   0
002360  000100                  EREG5:  .WORD   0
002362  000000                  EREG6:  .WORD   0
002364  000000                  EREG7:  .WORD   0

                                        ;#CONSTANTS
002366  052525  177777  125252  PREVAC: .WORD   ALTP,M1,ALTN,0  ; PREV CONTENTS OF FLOAT AC
002374  000000

                                        ;#MESSAGES FOR BEGIN PROGRAM/START OF PASS/ETC
002376  005015  005012  042115  BGNMES: .ASCII  <15><12><12><12>"MD-11-DQFPA-A..."
002404  030455  026461  050504
002412  050106  026501  027101
002420  027056
002422  042120  026520  030461          .ASCIZ  "PDP-11/6X F.P.U. BASIC INSTRUCTION TESTS"<15><12>
002430  033057  020130  027106
002436  027120  027125  041040
002444  051501  041511  044440
002452  051516  051124  041525
002460  044524  047117  052040
002466  051505  051524  005015
002474     000
002475     015  050012  051501  NWPAS1: .ASCIZ  <15><12>"PASS #"
002502  020123  000043
```

```
                                        .SBTTL  START OF PASS ROUTINE

                                                .EVEN                   ; START ON AN EVEN BOUNDARY

                                        ;;**********************************************************************
                                                .ENABL  AMA             ; BEGIN ASSEMBLING RELATIVE REFERENCES AS ABSOLUTE
                                        ;;**********************************************************************

002506                                  START:
                                        .SBTTL  INITIALIZE THE COMMON TAGS
                                        ;;CLEAR THE COMMON TAGS ($CMTAG) AREA
002506  012706  001100                          MOV     #$CMTAG,R6      ;;FIRST LOCATION TO BE CLEARED
002512  005026                                  CLR     (R6)+           ;;CLEAR MEMORY LOCATION
002514  022706  001146                          CMP     #$WR,R6 ;;DONE?
002520  001374                                  BNE     .-6             ;;LOOP BACK IF NO
002522  012706  001100                          MOV     #STACK,SP       ;;SETUP THE STACK POINTER
                                        ;;INITIALIZE A FEW VECTORS
002526  012737  023356  000020                  MOV     #$SCOPE,@#IOTVEC ;;IOT VECTOR FOR SCOPE ROUTINE
002534  012737  000340  000022                  MOV     #340,@#IOTVEC+2 ;;LEVEL 7
002542  012737  023634  000030                  MOV     #SERROR,@#EMTVEC ;;EMT VECTOR FOR ERROR ROUTINE
002550  012737  000340  000032                  MOV     #340,@#EMTVEC+2 ;;LEVEL 7
002556  012737  025270  000034                  MOV     #STRAP,@#TRAPVEC ;;TRAP VECTOR FOR TRAP CALLS
002564  012737  000340  000036                  MOV     #340,@#TRAPVEC+2;LEVEL 7
002572  012737  025336  000024                  MOV     #$PWRDN,@#PWRVEC ;;POWER FAILURE VECTOR
002600  012737  000340  000026                  MOV     #340,@#PWRVEC+2 ;;LEVEL 7
002606  013737  020460  020452                  MOV     $ENDCT,$EOPCT   ;;SETUP END-OF-PROGRAM COUNTER
002614  005037  001310                          CLR     $TIMES          ;;INITIALIZE NUMBER OF ITERATIONS
002620  005037  001312                          CLR     $ESCAPE         ;;CLEAR THE ESCAPE ON ERROR ADDRESS
002624  012737  000001  001122                  MOV     #1,$ERMAX       ;;ALLOW ONE ERROR PER TEST
002632  012737  002632  001110                  MOV     #.,$LPADR       ;;INITIALIZE THE LOOP ADDRESS FOR SCOPE
002640  012737  002640  001114                  MOV     #.,$LPERR       ;;SETUP THE ERROR LOOP ADDRESS
                                        ;;SIZE FOR A HARDWARE SWITCH REGISTER. IF NOT FOUND OR IT IS
                                        ;;EQUAL TO A "-1" SETUP FOR A SOFTWARE SWITCH REGISTER.
002646  013746  000004                          MOV     @#ERRVEC,-(SP)  ;;SAVE ERROR VECTOR
002652  012737  002706  000004                  MOV     #64$,@#ERRVEC   ;;SET UP ERROR VECTOR
002660  012737  177570  001146                  MOV     #DSWR,SWR       ;;SETUP FOR A HARDWARE SWICH REGISTER
002666  012737  177570  001150                  MOV     #DDISP,DISPLAY  ;;AND A HARDWARE DISPLAY REGISTER
002674  022777  177777  176244                  CMP     #-1,@SWR        ;;TRY TO REFERENCE HARDWARE SWR
002702  001012                                  BNE     66$             ;;BRANCH IF NO TIMEOUT TRAP OCCURRED
                                                                        ;;AND  THE HARDWARE SWR IS NOT = -1
002704  000403                                  BR      65$             ;;BRANCH IF NO TIMEOUT
002706  012716  002714          64$:            MOV     #65$,(SP)       ;;SET UP FOR TRAP RETURN
002712  000002                                  RTI
002714  012737  000176  001146  65$:            MOV     #SWREG,SWR      ;;POINT TO SOFTWARE SWR
002722  012737  000174  001150                  MOV     #DISPREG,DISPLAY
002730  012637  000004          66$:            MOV     (SP)+,@#ERRVEC  ;;RESTORE ERROR VECTOR

002734  005037  001332                          CLR     $PASS           ;;CLEAR PASS COUNT
002740  132737  000200  001345                  BITB    #APTSIZE,$ENVM  ;;TEST USER SIZE UNDER APT
002746  001403                                  BEQ     67$             ;;YES,USE NON-APT SWITCH
002750  012737  001346  001146                  MOV     #$SWREG,SWR     ;;NO,USE APT SWITCH REGISTER
002756                          67$:

                                        ; SET UP FPP UNEXPECTED TRAP CATCHER - - - - - - - - -
002756  012737  023316  000244                  MOV     #FPPILT,@#FPPVEC        ; NEW PC AT FPP TRAP
002764  005037  000246                          CLR     @#FPPVEC+2              ; NEW PS AT FPP TRAP
```

```
1317
1318  002770  104401  002376                  TYPE    ,BGNMES         ; ID MESSAGE AT START
1319
1320                                          ;////////////////////////////////////////////////////////////////////
1321                                          ;       MESSAGE ON WHETHER OR NOT HFP UNIT IS PRESENT
1322
1323  002774  076600  000022                  MED     ,RWHAMI                 ;WHAMI INTO R0
1324  003000  032700  000020                  BIT     #BIT04,R0               ;IS THERE A HFP UNIT ?
1325  003004  001403                          BEQ     70$                     ;NO, BR
1326  003006  104401  003022                  TYPE    ,68$                    ;INDICATE FP11-E PRESENT
1327  003012  000453                          BR      NEWPAS                  ;GO FOR SUBPASS INIT
1328  003014  104401  003062          70$:    TYPE    ,69$                    ;INDICATE NO FP11-E
1329  003020  000450                          BR      NEWPAS                  ;GO FOR SUBPASS INIT
1330
1331  003022  005015  020052  050106  68$:    .ASCIZ  <15><12>"* FP11-E HFP UNIT PRESENT *"<15><12>
1332  003030  030461  042455  044040
1333  003036  050106  052440  044516
1334  003044  020124  051120  051505
1335  003052  047105  020124  006452
1336  003060  000012
1337  003062  005015  020052  047516  69$:    .ASCIZ  <15><12>"* NO FP11-E HFP UNIT - ALL TESTS WFP ONLY *"<15><12>
1338  003070  043040  030520  026461
1339  003076  020105  043110  020120
1340  003104  047125  052111  026440
1341  003112  040440  046114  052040
1342  003120  051505  051524  053440
1343  003126  050106  047440  046116
1344  003134  020131  006452  000012
1345                                          .EVEN
1346
1347                                          ;////////////////////////////////////////////////////////////////////
1348
1349
1350                                          ;;******************************************************************
1351                                          ;       NEW PASS ENTERS HERE
1352                                          ;;******************************************************************
1353
1354  003142  012706  001100          NEWPAS: MOV     #STACK,SP               ;RESET STACK PTR
1355
1356  003146  032777  010000  175772          BIT     #BIT12,@SWR             ;INHIBIT STATUS TYPEOUTS ?
1357  003154  001011                          BNE     SUBPAS                  ;BR IF YES
1358
1359  003156  104401  002475                  TYPE    ,NWPAS1                 ;"PASS #"
1360  003162  013746  001332                  MOV     $PASS,-(SP)             ;PASS COUNT INTO ...
1361  003166  005216                          INC     (SP)                    ;  1-N RANGE
1362  003170  104403                          TYPOS                           ;TYPE OCTAL
1363  003172     006     000                  .BYTE   6,0                     ;  6 DIGITS, NO LEADING ZEROS
1364  003174  104401  001321                  TYPE    ,$CRLF                  ;END THE LINE
1365
1366
1367                                          ;;******************************************************************
1368                                          ;       NEW SUBPASS ENTERS HERE
1369                                          ;;******************************************************************
1370
1371  003200  076600  000022          SUBPAS: MED     ,RWHAMI                 ;GET WHAMI INTO R0
1372  003204  032700  000020                  BIT     #BIT04,R0               ;1=HFP PRESENT, 0=NO
```

```
1373  003210  001430                        BEQ     20$                     ;IF NO HFP, TEST WARM ONLY
1374
1375  003212  076600  000144                MED     ,RFLAG                  ;GET FLAGS INTO R0
1376
1377  003216  032777  000002  175722        BIT     #SW01,@SWR              ;SW01:  1=HFP OR WFP TEST ONLY
1378  003224  001413                        BEQ     1$                      ;       0=ALTERNATE HFP/WFP PEF PASS
1379
1380  003226  032777  000001  175712        BIT     #SW00,@SWR              ;SW00:  1=WFP ONLY
1381  003234  001403                        BEQ     2$                      ;       0=HFP ONLY
1382  003236  042700  010000                BIC     #BIT12,R0               ;CLEAR HFP ENABLE FLAG<5> FOR WFP
1383  003242  000402                        BR      3$                      ;
1384  003244  052700  010000          2$:   BIS     #BIT12,R0               ;SET HFP ENABLE FLAG<5> FOR HFP
1385  003250  076600  000344          3$:   MED     ,WFLAG                  ;REWRITE FLAGS
1386
1387  003254  032700  010000          1$:   BIT     #BIT12,R0               ;TEST WHO'S ENABLED: HOT, WARM
1388  003260  001404                        BEQ     20$                     ;SET APPROPRIATE HEADER:
1389
1390  003262  012737  025526  024100  19$:   MOV     #ASCHOT,HOTWRM          ;"HOT: "
1391  003270  000403                        BR      21$                     ;
1392  003272  012737  025534  024100  20$:   MOV     #ASCWRM,HOTWRM          ;"WARM: "
1393  003300  005037  001102          21$:   CLR     $TSTNM                  ;ALL DONE, RESET TEST NUMBER COUNTER
```

```
1394                                        ;;*****************************************************************
1395                                        ;*TEST 1        TEST OF FPS REGISTER BY RIPPLING A 1
1396                                        ;;*****************************************************************
1397  003304  000004                       TST1:   SCOPE
1398  003306  012700  000001                       MOV     #BIT00,R0       ; INITIAL PATTERN
1399  003312  012737  003320  001114               MOV     #1$,$LPERR      ; ERROR LOOPING RETURN
1400
1401  003320  010037  001170               1$:     MOV     R0,$REG0        ; EDIT PATTERN SO BIT 4 OF FPS ( FMM )
1402  003324  042737  000020  001170               BIC     #BIT04,$REG0    ;   WILL ALWAYS BE OFF DURING TEST
1403  003332  170137  001170                       LDFPS   $REG0           ; LOAD FPS
1404  003336  042737  030000  001170               BIC     #BIT13!BIT12,$REG0      ; BITS 13 AND 12 ALWAYS READ 0
1405  003344  170237  001172                       STFPS   $REG1           ; STORE FPS
1406  003350  023737  001170  001172               CMP     $REG0,$REG1     ; LOAD/STORE WORK ?
1407  003356  001401                               BEQ     2$              ;   YES
1408  003360  104001                               ERROR   1               ;   NO - ERROR RETURN
1409  003362  000241                       2$:     CLC                     ; ROTATE IN A 0
1410  003364  006100                               ROL     R0              ; SHIFT PATTERN 1 POSITION LEFT
1411  003366  103354                               BCC     1$              ; CONT IF NOT YET FINISHED
1412
1413
1414
1415
1416                                        ;;*****************************************************************
1417                                        ;*TEST 2        TEST OF FPS REGISTER BY RIPPLING A 0
1418                                        ;;*****************************************************************
1419  003370  000004                       TST2:   SCOPE
1420  003372  012700  177776                       MOV     #^CBIT00,R0     ; INITIAL PATTERN
1421  003376  012737  003404  001114               MOV     #1$,$LPERR      ; ERROR LOOPING RETURN
1422
1423  003404  010037  001170               1$:     MOV     R0,$REG0        ; EDIT PATTERN SO BIT 4 OF FPS ( FMM )
1424  003410  042737  000020  001170               BIC     #BIT04,$REG0    ;   WILL ALWAYS BE OFF DURING TEST
1425  003416  170137  001170                       LDFPS   $REG0           ; LOAD FPS
1426  003422  042737  030000  001170               BIC     #BIT13!BIT12,$REG0      ; BITS 13 AND 12 ALWAYS READ 0
1427  003430  170237  001172                       STFPS   $REG1           ; STORE FPS
1428  003434  023737  001170  001172               CMP     $REG0,$REG1     ; LOAD/STORE WORK ?
1429  003442  001401                               BEQ     2$              ;   YES
1430  003444  104002                               ERROR   2               ;   NO - ERROR RETURN
1431  003446  000261                       2$:     SEC                     ; ROTATE IN A 1
1432  003450  006100                               ROL     R0              ; SHIFT PATTERN 1 POSITION LEFT
1433  003452  103754                               BCS     1$              ; CONT IF NOT YET FINISHED
```

```
1434                                    ;;*****************************************************************
1435                                    ;*TEST 3         TEST OF SETF INSTRUCTION
1436                                    ;;*****************************************************************
1437  003454  000004                    TST3:   SCOPE
1438  003456  170127  147757                    LDFPS   #FPS1           ; INITIAL PATTERN
1439  003462  170001                            SETF                    ; TEST IT
1440  003464  170237  001170                    STFPS   $REG0           ; GET NEW FPS
1441  003470  022737  147557  001170            CMP     #FPS1&^CBIT07,$REG0     ; DID SETF CLEAR FPS BIT 7 ?
1442  003476  001401                            BEQ     1$              ;
1443  003500  104003                            ERROR   3               ;   NO - ERROR
1444  003502                            1$:                           ;   YES - NEXT TEST
1445
1446
1447
1448
1449
1450                                    ;;*****************************************************************
1451                                    ;*TEST 4         TEST OF SETD INSTRUCTION
1452                                    ;;*****************************************************************
1453  003502  000004                    TST4:   SCOPE
1454  003504  170127  000000                    LDFPS   #FPS0           ; INITIAL PATTERN
1455  003510  170011                            SETD                    ; TEST IT
1456  003512  170237  001170                    STFPS   $REG0           ; GET NEW FPS
1457  003516  022737  000200  001170            CMP     #FPS0!BIT07,$REG0       ; DID SETD SET FPS BIT 7 ?
1458  003524  001401                            BEQ     1$              ;
1459  003526  104004                            ERROR   4               ;   NO - ERROR
1460  003530                            1$:                           ;   YES - NEXT TEST
1461
1462
1463
1464
1465
1466                                    ;;*****************************************************************
1467                                    ;*TEST 5         TEST OF SETI INSTRUCTION
1468                                    ;;*****************************************************************
1469  003530  000004                    TST5:   SCOPE
1470  003532  170127  147757                    LDFPS   #FPS1           ; INITIAL PATTERN
1471  003536  170002                            SETI                    ; TEST IT
1472  003540  170237  001170                    STFPS   $REG0           ; GET NEW FPS
1473  003544  022737  147657  001170            CMP     #FPS1&^CBIT06,$REG0     ; DID SETI CLEAR FPS BIT 6 ?
1474  003552  001401                            BEQ     1$              ;
1475  003554  104005                            ERROR   5               ;   NO - ERROR
1476  003556                            1$:                           ;   YES - NEXT TEST
1477
1478
1479
1480
1481
1482                                    ;;*****************************************************************
1483                                    ;*TEST 6         TEST OF SETL INSTRUCTION
1484                                    ;;*****************************************************************
1485  003556  000004                    TST6:   SCOPE
1486  003560  170127  000000                    LDFPS   #FPS0           ; INITIAL PATTERN
1487  003564  170012                            SETL                    ; TEST IT
1488  003566  170237  001170                    STFPS   $REG0           ; GET NEW FPS
1489  003572  022737  000100  001170            CMP     #FPS0!BIT06,$REG0       ; DID SETL SET FPS BIT 6 ?
```

```
1490  003600  001401                    BEQ     1$              ;
1491  003602  104006                    ERROR   6               ;   NO - ERROR
1492  003604                    1$:                             ;   YES - NEXT TEST
1493
```

```
1494
1495                                    ;;*****************************************************************
1496                                            .DSABL  AMA             ; ASSEMBLE ALL REFERENCES AS THEY ARE PRINTED
1497                                    ;;*****************************************************************
1498
1499                                    ;;*****************************************************************
1500                                    ;*TEST 7         TEST OF LOAD-SRC MODE-0, STORE-DST MODE-7(PC)
1501                                    ;;*****************************************************************
1502  003604  000004                   TST7:   SCOPE
1503  003606  016767  000036  175354           MOV     ADDR1,$REG0     ; GET TEST PATTERN
1504
1505  003614  016705  175350                   MOV     $REG0,R5        ; DATA
1506  003620  170105                           LDFPS   R5              ; M0-R5
1507
1508  003622  012767  001172  175344           MOV     #$REG1,$REG2    ; ADDR(DEST)
1509  003630  170277  175340                   STFPS   @$REG2          ; M7-R7
1510
1511  003634  026767  175330  175330           CMP     $REG0,$REG1     ; LOAD/STORE WORK?
1512  003642  001401                           BEQ     64$             ;
1513  003644  104007                           ERROR   7               ; NOT EQUAL, SIGNAL ERROR
1514  003646                           64$:
1515
1516  003646  000401                           BR      TST10           ;;
1517
1518  003650  105252                   ADDR1:  .WORD   105252          ; TEST PATTERN
1519
1520
1521                                    ;;*****************************************************************
1522                                    ;*TEST 10        TEST OF LOAD-SRC MODE-1, STORE-DST MODE-3
1523                                    ;;*****************************************************************
1524  003652  000004                   TST10:  SCOPE
1525  003654  016767  000050  175306           MOV     ADDR2,$REG0     ; GET TEST PATTERN
1526
1527  003662  012703  001170                   MOV     #$REG0,R3       ; ADDR(DATA)
1528  003666  170113                           LDFPS   (R3)            ; M1-R3
1529
1530  003670  012767  001172  175276           MOV     #$REG1,$REG2    ; ADDR(DEST)
1531  003676  012705  001174                   MOV     #$REG2,R5       ; ADDR(ADDR(DEST))
1532  003702  170235                           STFPS   @(R5)+          ; M3-R5
1533
1534  003704  020527  001176                   CMP     R5,#$REG2+2     ; WAS DST ADDR REG INCRE RIGHT AMOUNT?
1535  003710  001401                           BEQ     64$             ;
1536  003712  104010                           ERROR   10              ; NOT EQUAL, SIGNAL ERROR
1537  003714                           64$:
1538
1539  003714  026767  175250  175250           CMP     $REG0,$REG1     ; LOAD/STORE WORK?
1540  003722  001401                           BEQ     65$             ;
1541  003724  104011                           ERROR   11              ; NOT EQUAL, SIGNAL ERROR
1542  003726                           65$:
1543
1544  003726  000401                           BR      TST11           ;;
1545
1546  003730  042505                   ADDR2:  .WORD   042505          ; TEST PATTERN
1547
1548
1549                                    ;;*****************************************************************
```

```
                                        ;*TEST 11       TEST OF LOAD-SRC MODE-2, STORE-DST MODE-5
                                        ;;*****************************************************************
003732  000004                          TST11:  SCOPE
003734  016767  000060  175226                  MOV     ADDR3,$REG0     ; GET TEST PATTERN

003742  012701  001170                          MOV     #$REG0,R1       ; ADDR(DATA)
003746  170121                          LDFPS   (R1)+           ; M2-R1

003750  012767  001172  175216          MOV     #$REG1,$REG2    ; ADDR(DEST)
003756  012703  001176                  MOV     #$REG2+2,R3     ; ADDR(ADDR(DEST)+2)
003762  170253                          STFPS   @-(R3)          ; M5-R3

003764  020127  001172                  CMP     R1,#$REG0+2     ; WAS SRC ADDR REG INCRE RIGHT AMOUNT?
003770  001401                          BEQ     64$             ;
003772  104012                          ERROR   12              ; NOT EQUAL, SIGNAL ERROR
003774                          64$:

003774  020327  001174                  CMP     R3,#$REG2       ; WAS DST ADDR REG DECRE RIGHT AMOUNT?
004000  001401                          BEQ     65$             ;
004002  104013                          ERROR   13              ; NOT EQUAL, SIGNAL ERROR
004004                          65$:

004004  026767  175160  175160          CMP     $REG0,$REG1     ; LOAD/STORE WORK?
004012  001401                          BEQ     66$             ;
004014  104014                          ERROR   14              ; NOT EQUAL, SIGNAL ERROR
004016                          66$:

004016  000401                          BR      TST12           ;;

004020  105252                  ADDR3:  .WORD   105252          ; TEST PATTERN


                                        ;;*****************************************************************
                                        ;*TEST 12       TEST OF LOAD-SRC MODE-3, STORE-DST MODE-6(PC)
                                        ;;*****************************************************************
004022  000004                  TST12:  SCOPE
004024  016767  000046  175136          MOV     ADDR4,$REG0     ; GET TEST PATTERN

004032  012767  001170  175134          MOV     #$REG0,$REG2    ; ADDR(DATA)
004040  012700  001174                  MOV     #$REG2,R0       ; ADDR(ADDR(DATA))
004044  170130                          LDFPS   @(R0)+          ; M3-R0

004046  1"0267  175120                  STFPS   $REG1           ; M6-R7

004052  020027  001176                  CMP     R0,#$REG2+2     ; WAS SRC ADDR REG INCRE RIGHT AMOUNT?
004056  001401                          BEQ     64$             ;
004060  104015                          ERROR   15              ; NOT EQUAL, SIGNAL ERROR
004062                          64$:

004062  026767  175102  175102          CMP     $REG0,$REG1     ; LOAD/STORE WORK?
004070  001401                          BEQ     65$             ;
004072  104016                          ERROR   16              ; NOT EQUAL, SIGNAL ERROR
004074                          65$:

004074  000401                          BR      TST13           ;;
```

```
1606  004076  042505                  ADDR4:  .WORD   042505          ; TEST PATTERN
1607
1608
1609                                  ;;*****************************************************************
1610                                  ;*TEST 13       TEST OF LOAD-SRC MODE-4, STORE-DST MODE-0
1611                                  ;;*****************************************************************
1612  004100  000004                  TST13:  SCOPE
1613  004102  016767  000042  175060          MOV     ADDR5,$REG0     ; GET TEST PATTERN
1614
1615  004110  012702  001172                   MOV     #$REG0+2,R2     ; ADDR(DATA+2)
1616  004114  170142                           LDFPS   -(R2)           ; M4-R2
1617
1618  004116  170200                           STFPS   R0              ; M0-R0
1619  004120  010067  175046                   MOV     R0,$REG1        ; DEST
1620
1621  004124  020227  001170                   CMP     R2,#$REG0       ; WAS SRC ADDR REG DECRE RIGHT AMOUNT?
1622  004130  001401                           BEQ     64$             ;
1623  004132  104017                           ERROR   17              ; NOT EQUAL, SIGNAL ERROR
1624  004134                          64$:
1625
1626  004134  026767  175030  175030          CMP     $REG0,$REG1     ; LOAD/STORE WORK?
1627  004142  001401                           BEQ     65$             ;
1628  004144  104020                           ERROR   20              ; NOT EQUAL, SIGNAL ERROR
1629  004146                          65$:
1630
1631  004146  000401                           BR      TST14           ;;
1632
1633  004150  105252                  ADDR5:  .WORD   105252          ; TEST PATTERN
1634
1635
1636                                  ;;*****************************************************************
1637                                  ;*TEST 14       TEST OF LOAD-SRC MODE-5, STORE-DST MODE-3(PC)
1638                                  ;;*****************************************************************
1639  004152  000004                  TST14:  SCOPE
1640  004154  016767  000046  175006          MOV     ADDR6,$REG0     ; GET TEST PATTERN
1641
1642  004162  012767  001170  175004           MOV     #$REG0,$REG2    ; ADDR(DATA)
1643  004170  012704  001176                   MOV     #$REG2+2,R4     ; ADDR(ADDR(DATA)+2)
1644  004174  170154                           LDFPS   @-(R4)          ; M5-R4
1645
1646  004176  170237  001172                   STFPS   @#$REG1         ; M3-R7
1647
1648  004202  020427  001174                   CMP     R4,#$REG2       ; WAS SRC ADDR REG DECRE RIGHT AMOUNT?
1649  004206  001401                           BEQ     64$             ;
1650  004210  104021                           ERROR   21              ; NOT EQUAL, SIGNAL ERROR
1651  004212                          64$:
1652
1653  004212  026767  174752  174752          CMP     $REG0,$REG1     ; LOAD/STORE WORK?
1654  004220  001401                           BEQ     65$             ;
1655  004222  104022                           ERROR   22              ; NOT EQUAL, SIGNAL ERROR
1656  004224                          65$:
1657
1658  004224  000401                           BR      TST15           ;;
1659
1660  004226  042505                  ADDR6:  .WORD   042505          ; TEST PATTERN
1661
```

```
                                        ;;*********************************************************************
                                        ;*TEST 15       TEST OF LOAD-SRC MODE-6, STORE-DST MODE-7
                                        ;;*********************************************************************
004230  000004                  TST15:  SCOPE
004232  016767  000044  174730          MOV     ADDR7,$REG0     ; GET TEST PATTERN

004240  012703  001254                  MOV     #$REG0+64,R3    ; ADDR(DATA)+64
004244  170163  177714                  LDFPS   -64(R3)         ; M6-R3

004250  012767  001172  174716          MOV     #$REG1,$REG2    ; ADDR(DEST)
004256  012701  001134                  MOV     #$REG2-40,R1    ; ADDR(ADDR(DEST))-40
004262  170271  000040                  STFPS   @40(R1)         ; M7-R1

004266  026767  174676  174676          CMP     $REG0,$REG1     ; LOAD/STORE WORK?
004274  001401                          BEQ     64$             ;
004276  104014                          ERROR   14              ; NOT EQUAL, SIGNAL ERROR
004300                          64$:

004300  000401                          BR      TST16           ;;

004302  105252                  ADDR7:  .WORD   105252          ; TEST PATTERN


                                        ;;*********************************************************************
                                        ;*TEST 16       TEST OF LOAD-SRC MODE-7, STORE-DST MODE-2(PC)
                                        ;;*********************************************************************
004304  000004                  TST16:  SCOPE
004306  016767  000064  174654          MOV     ADDR10,$REG0    ; GET TEST PATTERN

004314  012767  001170  174652          MOV     #$REG0,$REG2    ; ADDR(DATA)
004322  012702  001216                  MOV     #$REG2+22,R2    ; ADDR(ADDR(DATA))+22
004326  170172  177756                  LDFPS   @-22(R2)        ; M7-R2

004332  012767  104117  000002          MOV     #ERROR117,1$    ; SETUP ERROR CALL FOR WRONG INCREMENT
004340  170227                          STFPS   (PC)+           ; M2-R7
004342  000000                  1$:     .WORD   0               ; DEST
004344  000403                          BR      2$              ; THIS SHOULD BE NEXT INSTR AFTER FPP PC MODE 2
004346  104120                          ERROR   120             ; NOT HERE  (+4 INCRE)
004350  104121                          ERROR   121             ; OR HERE  (+6 INCRE)
004352  104122                          ERROR   122             ; OR HERE  (+10 INCRE)
004354  016767  177762  174610  2$:     MOV     1$,$REG1        ; GET DEST

004362  026767  174602  174602          CMP     $REG0,$REG1     ; LOAD/STORE WORK?
004370  001401                          BEQ     64$             ;
004372  104023                          ERROR   23              ; NOT EQUAL, SIGNAL ERROR
004374                          64$:

004374  000401                          BR      TST17           ;;

004376  042505                  ADDR10: .WORD   042505          ; TEST PATTERN


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

```
1718  004400  000004                    TST17:  SCOPE
1719  004402  016767  000056  174560            MOV     ADDR11,$REG0    ; GET TEST PATTERN
1720
1721  004410  016767  174554  000002            MOV     $REG0,1$        ; PUT DATA
1722  004416  170127                            LDFPS   (PC)+           ; M2-P7
1723  004420  000000                    1$:     .WORD   0               ; DATA
1724  004422  000403                            BR      2$              ; THIS SHOULD BE NEXT INSTR AFTER FPP PC MODE 2
1725  004424  104120                            ERROR   120             ; NOT HERE  (+4 INCRE)
1726  004426  104121                            ERROR   121             ; OR HERE  (+6 INCRE)
1727  004430  104122                            ERROR   122             ; OR HERE  (+10 INCRE)
1728
1729  004432  012703  001174            2$:     MOV     #$REG1+2,R3     ; ADDR(DEST+2)
1730  004436  170243                            STFPS   -(R3)           ; M4-R3
1731
1732  004440  020327  001172                    CMP     R3,#$REG1       ; WAS DST ADDR REG DECRE RIGHT AMOUNT?
1733  004444  001401                            BEQ     64$             ;
1734  004446  104024                            ERROR   24              ; NOT EQUAL, SIGNAL ERROR
1735  004450                            64$:
1736
1737  004450  026767  174514  174514            CMP     $REG0,$REG1     ; LOAD/STORE WORK?
1738  004456  001401                            BEQ     65$             ;
1739  004460  104025                            ERROR   25              ; NOT EQUAL, SIGNAL ERROR
1740  004462                            65$:
1741
1742  004462  000401                            BR      TST20           ;;
1743
1744  004464  104117                    ADDR11: .WORD   ERROR117                ; TEST PATTERN
1745
1746
1747                                    ;;**********************************************************************
1748                                    ;*TEST 20       TEST OF LOAD-SRC MODE-3(PC), STORE-DST MODE-6
1749                                    ;;**********************************************************************
1750  004466  000004                    TST20:  SCOPE
1751  004470  016767  000032  174472            MOV     ADDR12,$REG0    ; GET TEST PATTERN
1752
1753  004476  170137  001170                    LDFPS   @#$REG0         ; M3-R7
1754
1755  004502  012702  001144                    MOV     #$REG1-26,R2    ; ADDR(DEST)-26
1756  004506  170262  000026                    STFPS   26(R2)          ; M6-R2
1757
1758  004512  026767  174452  174452            CMP     $REG0,$REG1     ; LOAD/STORE WORK?
1759  004520  001401                            BEQ     64$             ;
1760  004522  104026                            ERROR   26              ; NOT EQUAL, SIGNAL ERROR
1761  004524                            64$:
1762
1763  004524  000401                            BR      TST21           ;;
1764
1765  004526  042505                    ADDR12: .WORD   042505          ; TEST PATTERN
1766
1767
1768                                    ;;**********************************************************************
1769                                    ;*TEST 21       TEST OF LOAD-SRC MODE-6(PC), STORE-DST MODE-1
1770                                    ;;**********************************************************************
1771  004530  000004                    TST21:  SCOPE
1772  004532  016767  000030  174430            MOV     ADDR13,$REG0    ; GET TEST PATTERN
1773
```

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

```
1816
1817                                    ;;*****************************************************************
1818                                            .ENABL  AMA             ; BEGIN ASSEMBLING RELATIVE REFERENCES AS ABSOLUTE
1819                                    ;;*****************************************************************
1820
1821                                    ;;*****************************************************************
1822                                    ;*TEST 23       TEST OF CFCC INSTRUCTION
1823                                    ;;*****************************************************************
1824  004646  000004                    TST23:  SCOPE
1825  004650  012700  000010                    MOV     #10,R0          ; NUMBER OF ENTRIES IN TABLE
1826  004654  012701  004760                    MOV     #OFCC,R1        ; START OF TABLE
1827  004660  012737  004666  001114            MOV     #1$,$LPERR      ; ERROR LOOPING RETURN
1828
1829  004666  011137  001170            1$:     MOV     (R1),$REG0      ; GET FIRST FCC PATTERN
1830  004672  042737  177760  001170            BIC     #CCONLY,$REG0   ;  BITS IN FCC POSITIONS ONLY
1831  004700  170137  001170                    LDFPS   $REG0           ; STORE IN FPS REGISTER
1832  004704  032711  000040                    BIT     #BIT05,(R1)     ; TEST WHETHER FLOATING A 1 OR 0
1833  004710  001002                            BNE     2$              ;
1834  004712  000257                            CCC                     ; FOR FLOAT A 1, START W/ CC = 0000
1835  004714  000401                            BR      3$              ;
1836  004716  000277                    2$:     SCC                     ; FOR FLOAT A 0, START W/ CC = 1111
1837  004720  170000                    3$:     CFCC                    ; COPY THE CONDITION CODES
1838  004722  013737  177776  001172            MOV     @#PSW,$REG1     ; GET CPU CC BITS
1839  004730  042737  177760  001172            BIC     #CCONLY,$REG1   ; CLEAR EXTRANEOUS BITS
1840  004736  023737  001170  001172            CMP     $REG0,$REG1     ; WERE THEY COPIED OK ?
1841  004744  001401                            BEQ     4$              ;
1842  004746  104030                            ERROR   30              ; NO - SIGNAL ERROR
1843  004750  062701  000002            4$:     ADD     #2,R1           ; INCRE R1 OUT OF ERROR LOOP
1844  004754  077034                            SOB     R0,1$           ; LOOP CONTROL
1845  004756  000410                            BR      TST24           ;;
1846
1847  004760  000001  000002  000004    OFCC:   .WORD   000001,000002,000004,000010     ; TABLE OF CC
1848  004766  000010
1849  004770  000056  000055  000053            .WORD   000056,000055,000053,000047     ;   TEST PATTERNS
1850  004776  000047
```

```
1851                                    ;;*****************************************************************
1852                                    ;*TEST 24       TEST FP AC0, D MODE, TEST PATTERN LDAD-1
1853                                    ;;*****************************************************************
1854  005000  000004                   TST24:  SCOPE
1855  005002  012704  020706                   MOV     #LDARD0,R4      ; PTR TO TESTING ROUTINE
1856  005006  012705  005020                   MOV     #LDAD1,R5       ; PTR TO TEST DATA SET
1857  005012  004737  020524                   JSR     PC,@#LDADT      ; GO TEST
1858
1859  005016  000413                           BR      TST25           ;;
1860
1861  005020                            LDAD1:  ; TEST DATA SET LDAD-1:
1862  005020  000000  000000  000000           .WORD   0,   0,   0,   0        ; PATTERN 1
1863  005026  000000
1864  005030  000000  000000  000000           .WORD   0,   0,   0,   0        ; PATTERN EXPECTED AFTER
1865  005036  000000
1866  005040  047653  047644                   .WORD   047653,047644   ; FPS:  BEFORE, AFTER
1867  005044  000000                           .WORD   000000          ; FEC AFTER ( 0 = N/A )
1868
1869                                    ;;*****************************************************************
1870                                    ;*TEST 25       TEST FP AC0, D MODE, TEST PATTERN LDAD-2
1871                                    ;;*****************************************************************
1872  005046  000004                   TST25:  SCOPE
1873  005050  012704  020706                   MOV     #LDARD0,R4      ; PTR TO TESTING ROUTINE
1874  005054  012705  005066                   MOV     #LDAD2,R5       ; PTR TO TEST DATA SET
1875  005060  004737  020524                   JSR     PC,@#LDADT      ; GO TEST
1876
1877  005064  000413                           BR      TST26           ;;
1878
1879  005066                            LDAD2:  ; TEST DATA SET LDAD-2:
1880  005066  177777  177777  177777           .WORD   M1,  M1,  M1,  M1       ; PATTERN 2
1881  005074  177777
1882  005076  177777  177777  177777           .WORD   M1,  M1,  M1,  M1       ; PATTERN EXPECTED AFTER
1883  005104  177777
1884  005106  047745  047750                   .WORD   047745,047750   ; FPS:  BEFORE, AFTER
1885  005112  000000                           .WORD   000000          ; FEC AFTER ( 0 = N/A )
1886
1887                                    ;;*****************************************************************
1888                                    ;*TEST 26       TEST FP AC0, D MODE, TEST PATTERN LDAD-3
1889                                    ;;*****************************************************************
1890  005114  000004                   TST26:  SCOPE
1891  005116  012704  020706                   MOV     #LDARD0,R4      ; PTR TO TESTING ROUTINE
1892  005122  012705  005134                   MOV     #LDAD3,R5       ; PTR TO TEST DATA SET
1893  005126  004737  020524                   JSR     PC,@#LDADT      ; GO TEST
1894
1895  005132  000413                           BR      TST27           ;;
1896
1897  005134                            LDAD3:  ; TEST DATA SET LDAD-3:
1898  005134  052525  052525  052525           .WORD   ALTP,ALTP,ALTP,ALTP     ; PATTERN 3
1899  005142  052525
1900  005144  052525  052525  052525           .WORD   ALTP,ALTP,ALTP,ALTP     ; PATTERN EXPECTED AFTER
1901  005152  052525
1902  005154  047652  047640                   .WORD   047652,047640   ; FPS:  BEFORE, AFTER
1903  005160  000000                           .WORD   000000          ; FEC AFTER ( 0 = N/A )
1904
1905                                    ;;*****************************************************************
1906                                    ;*TEST 27       TEST FP AC0, D MODE, TEST PATTERN LDAD-4
```

```
1907
1908  005162  000004                    ;;*****************************************************************
1909  005164  012704  020706            TST27:  SCOPE
1910  005170  012705  005202                    MOV     #LDARD0,R4      ; PTR TO TESTING ROUTINE
1911  005174  004737  020524                    MOV     #LDAD4,R5       ; PTR TO TEST DATA SET
1912                                            JSR     PC,@#LDADT      ; GO TEST
1913  005200  000413                           BR      TST30           ;;
1914
1915  005202                            LDAD4:  ; TEST DATA SET LDAD-4:
1916  005202  125252  125252  125252            .WORD   ALTN,ALTN,ALTN,ALTN     ; PATTERN 4
1917  005210  125252
1918  005212  125252  125252  125252            .WORD   ALTN,ALTN,ALTN,ALTN     ; PATTERN EXPECTED AFTER
1919  005220  125252
1920  005222  047704  047710                   .WORD   047704,047710   ; FPS:  BEFORE, AFTER
1921  005226  000000                           .WORD   000000          ; FEC AFTER ( 0 = N/A )
1922
1923                                    ;;*****************************************************************
1924                                    ;*TEST 30        TEST FP AC0, D MODE, TEST PATTERN LDAD-5
1925                                    ;;*****************************************************************
1926  005230  000004                    TST30:  SCOPE
1927  005232  012704  020706                    MOV     #LDARD0,R4      ; PTR TO TESTING ROUTINE
1928  005236  012705  005250                    MOV     #LDAD5,R5       ; PTR TO TEST DATA SET
1929  005242  004737  020524                    JSR     PC,@#LDADT      ; GO TEST
1930
1931  005246  000413                           BR      TST31           ;;
1932
1933  005250                            LDAD5:  ; TEST DATA SET LDAD-5:
1934  005250  100177  177777  177777            .WORD   ZX1MN,M1,M1,M1          ; PATTERN 5
1935  005256  177777
1936  005260  100177  177777  177777            .WORD   ZX1MN,M1,M1,M1          ; PATTERN EXPECTED AFTER
1937  005266  177777
1938  005270  043743  043754                   .WORD   043743,043754   ; FPS:  BEFORE, AFTER
1939  005274  000000                           .WORD   000000          ; FEC AFTER ( 0 = N/A )
1940
1941                                    ;;*****************************************************************
1942                                    ;*TEST 31        TEST FP AC0, D MODE, TEST PATTERN LDAD-6
1943                                    ;;*****************************************************************
1944  005276  000004                    TST31:  SCOPE
1945  005300  012704  020706                    MOV     #LDARD0,R4      ; PTR TO TESTING ROUTINE
1946  005304  012705  005316                    MOV     #LDAD6,R5       ; PTR TO TEST DATA SET
1947  005310  004737  020524                    JSR     PC,@#LDADT      ; GO TEST
1948
1949  005314  000413                           BR      TST32           ;;
1950
1951  005316                            LDAD6:  ; TEST DATA SET LDAD-6:
1952  005316  100000  000000  000000            .WORD   M0,  0,   0,   0        ; PATTERN 6
1953  005324  000000
1954  005326  052525  177777  125252            .WORD   ALTP,M1,  ALTN,0        ; PATTERN EXPECTED AFTER
1955  005334  000000
1956  005336  047602  147614                   .WORD   047602,147614   ; FPS:  BEFORE, AFTER
1957  005342  100014                           .WORD   100014          ; FEC AFTER ( 0 = N/A )
1958
1959                                    ;;*****************************************************************
1960                                    ;*TEST 32        TEST FP AC1, D MODE, TEST PATTERN LDAD-1
1961                                    ;;*****************************************************************
1962  005344  000004                    TST32:  SCOPE
```

```
FPU BASIC INSTR TESTS   MACY11 27(1006)  09-FEB-77  09:48  PAGE 44
DQFPAA.P11    09-FEB-77 09:42           T32     TEST FP AC1, D MODE, TEST PATTERN LDAD-1

 1963  005346  012704  020722                   MOV     #LDARD1,R4      ; PTR TO TESTING ROUTINE
 1964  005352  012705  005020                   MOV     #LDAD1,R5       ; PTR TO TEST DATA SET
 1965  005356  004737  020524                   JSR     PC,@#LDADT      ; GO TEST
 1966
 1967  005362  000400                           BR      TST33           ;;
 1968
 1969                                   ; ( FOR DATA SEE TEST OF FP REG AC0 WITH TEST PATTERN LDAD-1, ABOVE )
 1970
 1971                                   ;;**********************************************************************
 1972                                   ;*TEST 33       TEST FP AC1, D MODE, TEST PATTERN LDAD-2
 1973                                   ;;**********************************************************************
 1974  005364  000004                   TST33:  SCOPE
 1975  005366  012704  020722                   MOV     #LDARD1,R4      ; PTR TO TESTING ROUTINE
 1976  005372  012705  005066                   MOV     #LDAD2,R5       ; PTR TO TEST DATA SET
 1977  005376  004737  020524                   JSR     PC,@#LDADT      ; GO TEST
 1978
 1979  005402  000400                           BR      TST34           ;;
 1980
 1981                                   ; ( FOR DATA SEE TEST OF FP REG AC0 WITH TEST PATTERN LDAD-2, ABOVE )
 1982
 1983                                   ;;**********************************************************************
 1984                                   ;*TEST 34       TEST FP AC1, D MODE, TEST PATTERN LDAD-3
 1985                                   ;;**********************************************************************
 1986  005404  000004                   TST34:  SCOPE
 1987  005406  012704  020722                   MOV     #LDARD1,R4      ; PTR TO TESTING ROUTINE
 1988  005412  012705  005134                   MOV     #LDAD3,R5       ; PTR TO TEST DATA SET
 1989  005416  004737  020524                   JSR     PC,@#LDADT      ; GO TEST
 1990
 1991  005422  000400                           BR      TST35           ;;
 1992
 1993                                   ; ( FOR DATA SEE TEST OF FP REG AC0 WITH TEST PATTERN LDAD-3, ABOVE )
 1994
 1995                                   ;;**********************************************************************
 1996                                   ;*TEST 35       TEST FP AC1, D MODE, TEST PATTERN LDAD-4
 1997                                   ;;**********************************************************************
 1998  005424  000004                   TST35:  SCOPE
 1999  005426  012704  020722                   MOV     #LDARD1,R4      ; PTR TO TESTING ROUTINE
 2000  005432  012705  005202                   MOV     #LDAD4,R5       ; PTR TO TEST DATA SET
 2001  005436  004737  020524                   JSR     PC,@#LDADT      ; GO TEST
 2002
 2003  005442  000400                           BR      TST36           ;;
 2004
 2005                                   ; ( FOR DATA SEE TEST OF FP REG AC0 WITH TEST PATTERN LDAD-4, ABOVE )
 2006
 2007                                   ;;**********************************************************************
 2008                                   ;*TEST 36       TEST FP AC1, D MODE, TEST PATTERN LDAD-5
 2009                                   ;;**********************************************************************
 2010  005444  000004                   TST36:  SCOPE
 2011  005446  012704  020722                   MOV     #LDARD1,R4      ; PTR TO TESTING ROUTINE
 2012  005452  012705  005250                   MOV     #LDAD5,R5       ; PTR TO TEST DATA SET
 2013  005456  004737  020524                   JSR     PC,@#LDADT      ; GO TEST
 2014
 2015  005462  000400                           BR      TST37           ;;
 2016
 2017                                   ; ( FOR DATA SEE TEST OF FP REG AC0 WITH TEST PATTERN LDAD-5, ABOVE )
 2018
```

```
2019                                    ;;*****************************************************************
2020                                    ;*TEST 37        TEST FP AC1, D MODE, TEST PATTERN LDAD-6
2021                                    ;;*****************************************************************
2022  005464  000004                    TST37:  SCOPE
2023  005466  012704  020722                    MOV     #LDARD1,R4      ; PTR TO TESTING ROUTINE
2024  005472  012705  005316                    MOV     #LDAD6,R5       ; PTR TO TEST DATA SET
2025  005476  004737  020524                    JSR     PC,@#LDADT      ; GO TEST
2026
2027  005502  000400                            BR      TST40           ;;
2028
2029                                    ; ( FOR DATA SEE TEST OF FP REG AC0 WITH TEST PATTERN LDAD-6, ABOVE )
2030
2031
2032                                    ;;*****************************************************************
2033                                    ;*TEST 40        TEST FP AC2, D MODE, TEST PATTERN LDAD-1
2034                                    ;;*****************************************************************
2035  005504  000004                    TST40:  SCOPE
2036  005506  012704  020736                    MOV     #LDARD2,R4      ; PTR TO TESTING ROUTINE
2037  005512  012705  005020                    MOV     #LDAD1,R5       ; PTR TO TEST DATA SET
2038  005516  004737  020524                    JSR     PC,@#LDADT      ; GO TEST
2039
2040  005522  000400                            BR      TST41           ;;
2041
2042                                    ; ( FOR DATA SEE TEST OF FP REG AC0 WITH TEST PATTERN LDAD-1, ABOVE )
2043
2044                                    ;;*****************************************************************
2045                                    ;*TEST 41        TEST FP AC2, D MODE, TEST PATTERN LDAD-2
2046                                    ;;*****************************************************************
2047  005524  000004                    TST41:  SCOPE
2048  005526  012704  020736                    MOV     #LDARD2,R4      ; PTR TO TESTING ROUTINE
2049  005532  012705  005066                    MOV     #LDAD2,R5       ; PTR TO TEST DATA SET
2050  005536  004737  020524                    JSR     PC,@#LDADT      ; GO TEST
2051
2052  005542  000400                            BR      TST42           ;;
2053
2054                                    ; ( FOR DATA SEE TEST OF FP REG AC0 WITH TEST PATTERN LDAD-2, ABOVE )
2055
2056                                    ;;*****************************************************************
2057                                    ;*TEST 42        TEST FP AC2, D MODE, TEST PATTERN LDAD-3
2058                                    ;;*****************************************************************
2059  005544  000004                    TST42:  SCOPE
2060  005546  012704  020736                    MOV     #LDARD2,R4      ; PTR TO TESTING ROUTINE
2061  005552  012705  005134                    MOV     #LDAD3,R5       ; PTR TO TEST DATA SET
2062  005556  004737  020524                    JSR     PC,@#LDADT      ; GO TEST
2063
2064  005562  000400                            BR      TST43           ;;
2065
2066                                    ; ( FOR DATA SEE TEST OF FP REG AC0 WITH TEST PATTERN LDAD-3, ABOVE )
2067
2068                                    ;;*****************************************************************
2069                                    ;*TEST 43        TEST FP AC2, D MODE, TEST PATTERN LDAD-4
2070                                    ;;*****************************************************************
2071  005564  000004                    TST43:  SCOPE
2072  005566  012704  020736                    MOV     #LDARD2,R4      ; PTR TO TESTING ROUTINE
2073  005572  012705  005202                    MOV     #LDAD4,R5       ; PTR TO TEST DATA SET
2074  005576  004737  020524                    JSR     PC,@#LDADT      ; GO TEST
```

```
2075
2076  005602  000400                        BR      TST44           ;;
2077
2078                                  ; ( FOR DATA SEE TEST OF FP REG AC0 WITH TEST PATTERN LDAD-4, ABOVE )
2079
2080                                  ;;*****************************************************************
2081                                  ;*TEST 44       TEST FP AC2, D MODE, TEST PATTERN LDAD-5
2082                                  ;;*****************************************************************
2083  005604  000004                  TST44:  SCOPE
2084  005606  012704  020736                  MOV     #LDARD2,R4      ; PTR TO TESTING ROUTINE
2085  005612  012705  005250                  MOV     #LDAD5,R5       ; PTR TO TEST DATA SET
2086  005616  004737  020524                  JSR     PC,@#LDADT      ; GO TEST
2087
2088  005622  000400                        BR      TST45           ;;
2089
2090                                  ; ( FOR DATA SEE TEST OF FP REG AC0 WITH TEST PATTERN LDAD-5, ABOVE )
2091
2092                                  ;;*****************************************************************
2093                                  ;*TEST 45       TEST FP AC2, D MODE, TEST PATTERN LDAD-6
2094                                  ;;*****************************************************************
2095  005624  000004                  TST45:  SCOPE
2096  005626  012704  020736                  MOV     #LDARD2,R4      ; PTR TO TESTING ROUTINE
2097  005632  012705  005316                  MOV     #LDAD6,R5       ; PTR TO TEST DATA SET
2098  005636  004737  020524                  JSR     PC,@#LDADT      ; GO TEST
2099
2100  005642  000400                        BR      TST46           ;;
2101
2102                                  ; ( FOR DATA SEE TEST OF FP REG AC0 WITH TEST PATTERN LDAD-6, ABOVE )
2103
2104
2105                                  ;;*****************************************************************
2106                                  ;*TEST 46       TEST FP AC3, D MODE, TEST PATTERN LDAD-1
2107                                  ;;*****************************************************************
2108  005644  000004                  TST46:  SCOPE
2109  005646  012704  020752                  MOV     #LDARD3,R4      ; PTR TO TESTING ROUTINE
2110  005652  012705  005020                  MOV     #LDAD1,R5       ; PTR TO TEST DATA SET
2111  005656  004737  020524                  JSR     PC,@#LDADT      ; GO TEST
2112
2113  005662  000400                        BR      TST47           ;;
2114
2115                                  ; ( FOR DATA SEE TEST OF FP REG AC0 WITH TEST PATTERN LDAD-1, ABOVE )
2116
2117                                  ;;*****************************************************************
2118                                  ;*TEST 47       TEST FP AC3, D MODE, TEST PATTERN LDAD-2
2119                                  ;;*****************************************************************
2120  005664  000004                  TST47:  SCOPE
2121  005666  012704  020752                  MOV     #LDARD3,R4      ; PTR TO TESTING ROUTINE
2122  005672  012705  005066                  MOV     #LDAD2,R5       ; PTR TO TEST DATA SET
2123  005676  004737  020524                  JSR     PC,@#LDADT      ; GO TEST
2124
2125  005702  000400                        BR      TST50           ;;
2126
2127                                  ; ( FOR DATA SEE TEST OF FP REG AC0 WITH TEST PATTERN LDAD-2, ABOVE )
2128
2129                                  ;;*****************************************************************
2130                                  ;*TEST 50       TEST FP AC3, D MODE, TEST PATTERN LDAD-3
```

```
2131                                    ;;*****************************************************************
2132  005704  000004                    TST50:  SCOPE
2133  005706  012704  020752                    MOV     #LDARD3,R4      ; PTR TO TESTING ROUTINE
2134  005712  012705  005134                    MOV     #LDAD3,R5       ; PTR TO TEST DATA SET
2135  005716  004737  020524                    JSR     PC,@#LDADT      ; GO TEST
2136
2137  005722  000400                          BR      TST51           ;;
2138
2139                                    ; ( FOR DATA SEE TEST OF FP REG AC0 WITH TEST PATTERN LDAD-3, ABOVE )
2140
2141                                    ;;*****************************************************************
2142                                    ;*TEST 51        TEST FP AC3, D MODE, TEST PATTERN LDAD-4
2143                                    ;;*****************************************************************
2144  005724  000004                    TST51:  SCOPE
2145  005726  012704  020752                    MOV     #LDARD3,R4      ; PTR TO TESTING ROUTINE
2146  005732  012705  005202                    MOV     #LDAD4,R5       ; PTR TO TEST DATA SET
2147  005736  004737  020524                    JSR     PC,@#LDADT      ; GO TEST
2148
2149  005742  000400                          BR      TST52           ;;
2150
2151                                    ; ( FOR DATA SEE TEST OF FP REG AC0 WITH TEST PATTERN LDAD-4, ABOVE )
2152
2153                                    ;;*****************************************************************
2154                                    ;*TEST 52        TEST FP AC3, D MODE, TEST PATTERN LDAD-5
2155                                    ;;*****************************************************************
2156  005744  000004                    TST52:  SCOPE
2157  005746  012704  020752                    MOV     #LDARD3,R4      ; PTR TO TESTING ROUTINE
2158  005752  012705  005250                    MOV     #LDAD5,R5       ; PTR TO TEST DATA SET
2159  005756  004737  020524                    JSR     PC,@#LDADT      ; GO TEST
2160
2161  005762  000400                          BR      TST53           ;;
2162
2163                                    ; ( FOR DATA SEE TEST OF FP REG AC0 WITH TEST PATTERN LDAD-5, ABOVE )
2164
2165                                    ;;*****************************************************************
2166                                    ;*TEST 53        TEST FP AC3, D MODE, TEST PATTERN LDAD-6
2167                                    ;;*****************************************************************
2168  005764  000004                    TST53:  SCOPE
2169  005766  012704  020752                    MOV     #LDARD3,R4      ; PTR TO TESTING ROUTINE
2170  005772  012705  005316                    MOV     #LDAD6,R5       ; PTR TO TEST DATA SET
2171  005776  004737  020524                    JSR     PC,@#LDADT      ; GO TEST
2172
2173  006002  000400                          BR      TST54           ;;
2174
2175                                    ; ( FOR DATA SEE TEST OF FP REG AC0 WITH TEST PATTERN LDAD-6, ABOVE )
2176
2177
2178                                    ;;*****************************************************************
2179                                    ;*TEST 54        TEST FP AC4, D MODE, TEST PATTERN LDAD-1
2180                                    ;;*****************************************************************
2181  006004  000004                    TST54:  SCOPE
2182  006006  012704  020766                    MOV     #LDARD4,R4      ; PTR TO TESTING ROUTINE
2183  006012  012705  005020                    MOV     #LDAD1,R5       ; PTR TO TEST DATA SET
2184  006016  004737  020524                    JSR     PC,@#LDADT      ; GO TEST
2185
2186  006022  000400                          BR      TST55           ;;
```

```
                                        ; ( 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
                                        ;;*****************************************************************
006024  000004                          TST55:  SCOPE
006026  012704  020766                          MOV     #LDARD4,R4      ; PTR TO TESTING ROUTINE
006032  012705  005066                          MOV     #LDAD2,R5       ; PTR TO TEST DATA SET
006036  004737  020524                          JSR     PC,@#LDADT      ; GO TEST

006042  000400                          TST56:  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
                                        ;;*****************************************************************
006044  000004                          TST56:  SCOPE
006046  012704  020766                          MOV     #LDARD4,R4      ; PTR TO TESTING ROUTINE
006052  012705  005134                          MOV     #LDAD3,R5       ; PTR TO TEST DATA SET
006056  004737  020524                          JSR     PC,@#LDADT      ; GO TEST

006062  000400                          TST57:  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
                                        ;;*****************************************************************
006064  000004                          TST57:  SCOPE
006066  012704  020766                          MOV     #LDARD4,R4      ; PTR TO TESTING ROUTINE
006072  012705  005202                          MOV     #LDAD4,R5       ; PTR TO TEST DATA SET
006076  004737  020524                          JSR     PC,@#LDADT      ; GO TEST

006102  000400                          TST60:  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
                                        ;;*****************************************************************
006104  000004                          TST60:  SCOPE
006106  012704  020766                          MOV     #LDARD4,R4      ; PTR TO TESTING ROUTINE
006112  012705  005250                          MOV     #LDAD5,R5       ; PTR TO TEST DATA SET
006116  004737  020524                          JSR     PC,@#LDADT      ; GO TEST

006122  000400                          TST61:  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
                                        ;;*****************************************************************
006124  000004                          TST61:  SCOPE
```

```
                                T61     TEST FP AC5, D MODE, TEST PATTERN LDAD-1

2243  006126  012704  021006                MOV     #LDARD5,R4      ; PTR TO TESTING ROUTINE
2244  006132  012705  005020                MOV     #LDAD1,R5       ; PTR TO TEST DATA SET
2245  006136  004737  020524                JSR     PC,@#LDADT      ; GO TEST
2246
2247  006142  000400                        BR      TST62           ;;
2248
2249                                  ; ( FOR DATA SEE TEST OF FP REG AC0 WITH TEST PATTERN LDAD-1, ABOVE )
2250
2251                                  ;;**********************************************************************
2252                                  ;*TEST 62        TEST FP AC5, D MODE, TEST PATTERN LDAD-2
2253                                  ;;**********************************************************************
2254  006144  000004                TST62:  SCOPE
2255  006146  012704  021006                MOV     #LDARD5,R4      ; PTR TO TESTING ROUTINE
2256  006152  012705  005066                MOV     #LDAD2,R5       ; PTR TO TEST DATA SET
2257  006156  004737  020524                JSR     PC,@#LDADT      ; GO TEST
2258
2259  006162  000400                        BR      TST63           ;;
2260
2261                                  ; ( FOR DATA SEE TEST OF FP REG AC0 WITH TEST PATTERN LDAD-2, ABOVE )
2262
2263                                  ;;**********************************************************************
2264                                  ;*TEST 63        TEST FP AC5, D MODE, TEST PATTERN LDAD-3
2265                                  ;;**********************************************************************
2266  006164  000004                TST63:  SCOPE
2267  006166  012704  021006                MOV     #LDARD5,R4      ; PTR TO TESTING ROUTINE
2268  006172  012705  005134                MOV     #LDAD3,R5       ; PTR TO TEST DATA SET
2269  006176  004737  020524                JSR     PC,@#LDADT      ; GO TEST
2270
2271  006202  000400                        BR      TST64           ;;
2272
2273                                  ; ( FOR DATA SEE TEST OF FP REG AC0 WITH TEST PATTERN LDAD-3, ABOVE )
2274
2275                                  ;;**********************************************************************
2276                                  ;*TEST 64        TEST FP AC5, D MODE, TEST PATTERN LDAD-4
2277                                  ;;**********************************************************************
2278  006204  000004                TST64:  SCOPE
2279  006206  012704  021006                MOV     #LDARD5,R4      ; PTR TO TESTING ROUTINE
2280  006212  012705  005202                MOV     #LDAD4,R5       ; PTR TO TEST DATA SET
2281  006216  004737  020524                JSR     PC,@#LDADT      ; GO TEST
2282
2283  006222  000400                        BR      TST65           ;;
2284
2285                                  ; ( FOR DATA SEE TEST OF FP REG AC0 WITH TEST PATTERN LDAD-4, ABOVE )
2286
2287                                  ;;**********************************************************************
2288                                  ;*TEST 65        TEST FP AC5, D MODE, TEST PATTERN LDAD-5
2289                                  ;;**********************************************************************
2290  006224  000004                TST65:  SCOPE
2291  006226  012704  021006                MOV     #LDARD5,R4      ; PTR TO TESTING ROUTINE
2292  006232  012705  005250                MOV     #LDAD5,R5       ; PTR TO TEST DATA SET
2293  006236  004737  020524                JSR     PC,@#LDADT      ; GO TEST
2294
2295  006242  000400                        BR      TST66           ;;
2296
2297                                  ; ( FOR DATA SEE TEST OF FP REG AC0 WITH TEST PATTERN LDAD-5, ABOVE )
2298
```

```
2299
2300
```

```
2301                                    ;;*****************************************************************
2302                                    ;*TEST 66       TEST FP AC0, F MODE, TEST PATTERN LDAF-1
2303                                    ;;*****************************************************************
2304  006244  000004                   TST66:  SCOPE
2305  006246  012704  021206                   MOV     #LDARF0,R4      ; PTR TO TESTING ROUTINE
2306  006252  012705  006264                   MOV     #LDAF1,R5       ; PTR TO TEST DATA SET
2307  006256  004737  021026                   JSR     PC,@#LDAFT      ; GO TEST
2308
2309  006262  000415                           BR      TST67           ;;
2310
2311  006264                           LDAF1:  ; TEST DATA SET LDAF-1:
2312  006264  000000  000000  000000           .WORD   0,0,0,0                 ; LDD PATTERN
2313  006272  000000
2314  006274  177777  177777                   .WORD   M1,M1                   ; LDF PATTERN
2315  006300  177777  177777  000000           .WORD   M1,M1,0,0               ; STD EXPECTED PATTERN
2316  006306  000000
2317  006310  047547  047750                   .WORD   047547,047750   ; FPS:  BEFORE, AFTER
2318  006314  000000                           .WORD   000000          ; FEC AFTER ( 0 = N/A )
2319
2320                                    ;;*****************************************************************
2321                                    ;*TEST 67       TEST FP AC0, F MODE, TEST PATTERN LDAF-2
2322                                    ;;*****************************************************************
2323  006316  000004                   TST67:  SCOPE
2324  006320  012704  021206                   MOV     #LDARF0,R4      ; PTR TO TESTING ROUTINE
2325  006324  012705  006336                   MOV     #LDAF2,R5       ; PTR TO TEST DATA SET
2326  006330  004737  021026                   JSR     PC,@#LDAFT      ; GO TEST
2327
2328  006334  000415                           BR      TST70           ;;
2329
2330  006336                           LDAF2:  ; TEST DATA SET LDAF-2:
2331  006336  177777  177777  177777           .WORD   M1,M1,M1,M1             ; LDD PATTERN
2332  006344  177777
2333  006346  000000  000000                   .WORD   0,0                     ; LDF PATTERN
2334  006352  000000  000000  177777           .WORD   0,0,M1,M1               ; STD EXPECTED PATTERN
2335  006360  177777
2336  006362  047413  047604                   .WORD   047413,047604   ; FPS:  BEFORE, AFTER
2337  006366  000000                           .WORD   000000          ; FEC AFTER ( 0 = N/A )
2338
2339                                    ;;*****************************************************************
2340                                    ;*TEST 70       TEST FP AC0, F MODE, TEST PATTERN LDAF-3
2341                                    ;;*****************************************************************
2342  006370  000004                   TST70:  SCOPE
2343  006372  012704  021206                   MOV     #LDARF0,R4      ; PTR TO TESTING ROUTINE
2344  006376  012705  006410                   MOV     #LDAF3,R5       ; PTR TO TEST DATA SET
2345  006402  004737  021026                   JSR     PC,@#LDAFT      ; GO TEST
2346
2347  006406  000415                           BR      TST71           ;;
2348
2349  006410                           LDAF3:  ; TEST DATA SET LDAF-3:
2350  006410  052525  052525  052525           .WORD   ALTP,ALTP,ALTP,ALTP     ; LDD PATTERN
2351  006416  052525
2352  006420  125252  125252                   .WORD   ALTN,ALTN               ; LDF PATTERN
2353  006424  125252  125252  052525           .WORD   ALTN,ALTN,ALTP,ALTP     ; STD EXPECTED PATTERN
2354  006432  052525
2355  006434  047547  047750                   .WORD   047547,047750   ; FPS:  BEFORE, AFTER
2356  006440  000000                           .WORD   000000          ; FEC AFTER ( 0 = N/A )
```

```
                                        ;;*****************************************************************
                                        ;*TEST 71        TEST FP AC0, F MODE, TEST PATTERN LDAF-4
                                        ;;*****************************************************************
006442  000004                  TST71:  SCOPE
006444  012704  021206                  MOV     #LDARF0,R4      ; PTR TO TESTING ROUTINE
006450  012705  006462                  MOV     #LDAF4,R5       ; PTR TO TEST DATA SET
006454  004737  021026                  JSR     PC,@#LDAFT      ; GO TEST

006460  000415                          BR      TST72           ;;

006462                          LDAF4:  ; TEST DATA SET LDAF-4:
006462  125252  125252  125252          .WORD   ALTN,ALTN,ALTN,ALTN     ; LDD PATTERN
006470  125252
006472  052525  052525                  .WORD   ALTP,ALTP               ; LDF PATTERN
006476  052525  052525  125252          .WORD   ALTP,ALTP,ALTN,ALTN     ; STD EXPECTED PATTERN
006504  125252
006506  047417  047600                  .WORD   047417,047600   ; FPS:  BEFORE, AFTER
006512  000000                          .WORD   000000          ; FEC AFTER ( 0 = N/A )

                                        ;;*****************************************************************
                                        ;*TEST 72        TEST FP AC0, F MODE, TEST PATTERN LDAF-5
                                        ;;*****************************************************************
006514  000004                  TST72:  SCOPE
006516  012704  021206                  MOV     #LDARF0,R4      ; PTR TO TESTING ROUTINE
006522  012705  006534                  MOV     #LDAF5,R5       ; PTR TO TEST DATA SET
006526  004737  021026                  JSR     PC,@#LDAFT      ; GO TEST

006532  000415                          BR      TST73           ;;

006534                          LDAF5:  ; TEST DATA SET LDAF-5:
006534  177777  177777  177777          .WORD   M1,M1,M1,M1             ; LDD PATTERN
006542  177777
006544  100000  000000                  .WORD   M0,00                   ; LDF PATTERN
006550  100000  000000  177777          .WORD   M0,00,M1,M1             ; STD EXPECTED PATTERN
006556  177777
006560  043443  043654                  .WORD   043443,043654   ; FPS:  BEFORE, AFTER
006564  000000                          .WORD   000000          ; FEC AFTER ( 0 = N/A )

                                        ;;*****************************************************************
                                        ;*TEST 73        TEST FP AC0, F MODE, TEST PATTERN LDAF-6
                                        ;;*****************************************************************
006566  000004                  TST73:  SCOPE
006570  012704  021206                  MOV     #LDARF0,R4      ; PTR TO TESTING ROUTINE
006574  012705  006606                  MOV     #LDAF6,R5       ; PTR TO TEST DATA SET
006600  004737  021026                  JSR     PC,@#LDAFT      ; GO TEST

006604  000415                          BR      TST74           ;;

006606                          LDAF6:  ; TEST DATA SET LDAF-6:
006606  052525  000000  125252          .WORD   ALTP,0,ALTN,M1          ; LDD PATTERN
006614  177777
006616  100000  000000                  .WORD   M0,0                    ; LDF PATTERN
006622  052525  000000  125252          .WORD   ALTP,0,ALTN,M1          ; STD EXPECTED PATTERN
006630  177777
006632  047543  147754                  .WORD   047543,147754   ; FPS:  BEFORE, AFTER
```

```
                                T73     TEST FP AC0, F MODE, TEST PATTERN LDAF-6

002413  006636  100014                        .WORD   100014          ; FEC AFTER ( 0 = N/A )
002414
002415
002416                          ;;*****************************************************************
002417                          ;*TEST 74       TEST FP AC1, F MODE, TEST PATTERN LDAF-1
002418                          ;*****************************************************************
002419  006640  000004         TST74:  SCOPE
002420  006642  012704  021230        MOV     #LDARF1,R4      ; PTR TO TESTING ROUTINE
002421  006646  012705  006264        MOV     #LDAF1,R5       ; PTR TO TEST DATA SET
002422  006652  004737  021026        JSR     PC,@#LDAFT      ; GO TEST
002423
002424  006656  000400                BR      TST75           ;;
002425
002426                          ; ( FOR DATA SEE TEST OF FP REG AC0 WITH TEST PATTERN LDAF-1, ABOVE )
002427
002428                          ;;*****************************************************************
002429                          ;*TEST 75       TEST FP AC1, F MODE, TEST PATTERN LDAF-2
002430                          ;*****************************************************************
002431  006660  000004         TST75:  SCOPE
002432  006662  012704  021230        MOV     #LDARF1,R4      ; PTR TO TESTING ROUTINE
002433  006666  012705  006336        MOV     #LDAF2,R5       ; PTR TO TEST DATA SET
002434  006672  004737  021026        JSR     PC,@#LDAFT      ; GO TEST
002435
002436  006676  000400                BR      TST76           ;;
002437
002438                          ; ( FOR DATA SEE TEST OF FP REG AC0 WITH TEST PATTERN LDAF-2, ABOVE )
002439
002440                          ;;*****************************************************************
002441                          ;*TEST 76       TEST FP AC1, F MODE, TEST PATTERN LDAF-3
002442                          ;*****************************************************************
002443  006700  000004         TST76:  SCOPE
002444  006702  012704  021230        MOV     #LDARF1,R4      ; PTR TO TESTING ROUTINE
002445  006706  012705  006410        MOV     #LDAF3,R5       ; PTR TO TEST DATA SET
002446  006712  004737  021026        JSR     PC,@#LDAFT      ; GO TEST
002447
002448  006716  000400                BR      TST77           ;;
002449
002450                          ; ( FOR DATA SEE TEST OF FP REG AC0 WITH TEST PATTERN LDAF-3, ABOVE )
002451
002452                          ;;*****************************************************************
002453                          ;*TEST 77       TEST FP AC1, F MODE, TEST PATTERN LDAF-4
002454                          ;*****************************************************************
002455  006720  000004         TST77:  SCOPE
002456  006722  012704  021230        MOV     #LDARF1,R4      ; PTR TO TESTING ROUTINE
002457  006726  012705  006462        MOV     #LDAF4,R5       ; PTR TO TEST DATA SET
002458  006732  004737  021026        JSR     PC,@#LDAFT      ; GO TEST
002459
002460  006736  000400                BR      TST100          ;;
002461
002462                          ; ( FOR DATA SEE TEST OF FP REG AC0 WITH TEST PATTERN LDAF-4, ABOVE )
002463
002464                          ;;*****************************************************************
002465                          ;*TEST 100      TEST FP AC1, F MODE, TEST PATTERN LDAF-5
002466                          ;*****************************************************************
002467  006740  000004         TST100: SCOPE
002468  006742  012704  021230        MOV     #LDARF1,R4      ; PTR TO TESTING ROUTINE
```

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

2469  006746  012705  006534                MOV     #LDAF5,R5       ; PTR TO TEST DATA SET
2470  006752  004737  021026                JSR     PC,@#LDAFT      ; GO TEST
2471
2472  006756  000400                        BR      TST101          ;;
2473
2474                                ; ( FOR DATA SEE TEST OF FP REG AC0 WITH TEST PATTERN LDAF-5, ABOVE )
2475
2476                                ;;*****************************************************************
2477                                ;*TEST 101       TEST FP AC1, F MODE, TEST PATTERN LDAF-6
2478                                ;;*****************************************************************
2479  006760  000004               TST101: SCOPE
2480  006762  012704  021230                MOV     #LDARF1,R4      ; PTR TO TESTING ROUTINE
2481  006766  012705  006606                MOV     #LDAF6,R5       ; PTR TO TEST DATA SET
2482  006772  004737  021026                JSR     PC,@#LDAFT      ; GO TEST
2483
2484  006776  000400                        BR      TST102          ;;
2485
2486                                ; ( FOR DATA SEE TEST OF FP REG AC0 WITH TEST PATTERN LDAF-6, ABOVE )
2487
2488
2489                                ;;*****************************************************************
2490                                ;*TEST 102       TEST FP AC2, F MODE, TEST PATTERN LDAF-1
2491                                ;;*****************************************************************
2492  007000  000004               TST102: SCOPE
2493  007002  012704  021252                MOV     #LDARF2,R4      ; PTR TO TESTING ROUTINE
2494  007006  012705  006264                MOV     #LDAF1,R5       ; PTR TO TEST DATA SET
2495  007012  004737  021026                JSR     PC,@#LDAFT      ; GO TEST
2496
2497  007016  000400                        BR      TST103          ;;
2498
2499                                ; ( FOR DATA SEE TEST OF FP REG AC0 WITH TEST PATTERN LDAF-1, ABOVE )
2500
2501                                ;;*****************************************************************
2502                                ;*TEST 103       TEST FP AC2, F MODE, TEST PATTERN LDAF-2
2503                                ;;*****************************************************************
2504  007020  000004               TST103: SCOPE
2505  007022  012704  021252                MOV     #LDARF2,R4      ; PTR TO TESTING ROUTINE
2506  007026  012705  006336                MOV     #LDAF2,R5       ; PTR TO TEST DATA SET
2507  007032  004737  021026                JSR     PC,@#LDAFT      ; GO TEST
2508
2509  007036  000400                        BR      TST104          ;;
2510
2511                                ; ( FOR DATA SEE TEST OF FP REG AC0 WITH TEST PATTERN LDAF-2, ABOVE )
2512
2513                                ;;*****************************************************************
2514                                ;*TEST 104       TEST FP AC2, F MODE, TEST PATTERN LDAF-3
2515                                ;;*****************************************************************
2516  007040  000004               TST104: SCOPE
2517  007042  012704  021252                MOV     #LDARF2,R4      ; PTR TO TESTING ROUTINE
2518  007046  012705  006410                MOV     #LDAF3,R5       ; PTR TO TEST DATA SET
2519  007052  004737  021026                JSR     PC,@#LDAFT      ; GO TEST
2520
2521  007056  000400                        BR      TST105          ;;
2522
2523                                ; ( FOR DATA SEE TEST OF FP REG AC0 WITH TEST PATTERN LDAF-3, ABOVE )
2524
```

```
                                ;;*****************************************************************
                                ;*TEST 105       TEST FP AC2, F MODE, TEST PATTERN LDAF-4
                                ;;*****************************************************************
007060  000004                  TST105: SCOPE
007062  012704  021252                  MOV     #LDARF2,R4      ; PTR TO TESTING ROUTINE
007066  012705  006462                  MOV     #LDAF4,R5       ; PTR TO TEST DATA SET
007072  004737  021026                  JSR     PC,@#LDAFT      ; GO TEST
                
007076  000400                          BR      TST106          ;;

                                ; ( FOR DATA SEE TEST OF FP REG AC0 WITH TEST PATTERN LDAF-4, ABOVE )

                                ;;*****************************************************************
                                ;*TEST 106       TEST FP AC2, F MODE, TEST PATTERN LDAF-5
                                ;;*****************************************************************
007100  000004                  TST106: SCOPE
007102  012704  021252                  MOV     #LDARF2,R4      ; PTR TO TESTING ROUTINE
007106  012705  006534                  MOV     #LDAF5,R5       ; PTR TO TEST DATA SET
007112  004737  021026                  JSR     PC,@#LDAFT      ; GO TEST
                
007116  000400                          BR      TST107          ;;

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

                                ;;*****************************************************************
                                ;*TEST 107       TEST FP AC2, F MODE, TEST PATTERN LDAF-6
                                ;;*****************************************************************
007120  000004                  TST107: SCOPE
007122  012704  021252                  MOV     #LDARF2,R4      ; PTR TO TESTING ROUTINE
007126  012705  006606                  MOV     #LDAF6,R5       ; PTR TO TEST DATA SET
007132  004737  021026                  JSR     PC,@#LDAFT      ; GO TEST
                
007136  000400                          BR      TST110          ;;

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


                                ;;*****************************************************************
                                ;*TEST 110       TEST FP AC3, F MODE, TEST PATTERN LDAF-1
                                ;;*****************************************************************
007140  000004                  TST110: SCOPE
007142  012704  021274                  MOV     #LDARF3,R4      ; PTR TO TESTING ROUTINE
007146  012705  006264                  MOV     #LDAF1,R5       ; PTR TO TEST DATA SET
007152  004737  021026                  JSR     PC,@#LDAFT      ; GO TEST
                
007156  000400                          BR      TST111          ;;

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

                                ;;*****************************************************************
                                ;*TEST 111       TEST FP AC3, F MODE, TEST PATTERN LDAF-2
                                ;;*****************************************************************
007160  000004                  TST111: SCOPE
007162  012704  021274                  MOV     #LDARF3,R4      ; PTR TO TESTING ROUTINE
007166  012705  006336                  MOV     #LDAF2,R5       ; PTR TO TEST DATA SET
007172  004737  021026                  JSR     PC,@#LDAFT      ; GO TEST
```

```
                                        T111    TEST FP AC3, F MODE, TEST PATTERN LDAF-2

2581
2582  007176  000400                            BR      TST112          ;;
2583
2584                                  ; ( FOR DATA SEE TEST OF FP REG AC0 WITH TEST PATTERN LDAF-2, ABOVE )
2585
2586                                  ;;*****************************************************************
2587                                  ;*TEST 112      TEST FP AC3, F MODE, TEST PATTERN LDAF-3
2588                                  ;;*****************************************************************
2589  007200  000004                  TST112: SCOPE
2590  007202  012704  021274                  MOV     #LDARF3,R4      ; PTR TO TESTING ROUTINE
2591  007206  012705  006410                  MOV     #LDAF3,R5       ; PTR TO TEST DATA SET
2592  007212  004737  021026                  JSR     PC,@#LDAFT      ; GO TEST
2593
2594  007216  000400                          BR      TST113          ;;
2595
2596                                  ; ( FOR DATA SEE TEST OF FP REG AC0 WITH TEST PATTERN LDAF-3, ABOVE )
2597
2598                                  ;;*****************************************************************
2599                                  ;*TEST 113      TEST FP AC3, F MODE, TEST PATTERN LDAF-4
2600                                  ;;*****************************************************************
2601  007220  000004                  TST113: SCOPE
2602  007222  012704  021274                  MOV     #LDARF3,R4      ; PTR TO TESTING ROUTINE
2603  007226  012705  006462                  MOV     #LDAF4,R5       ; PTR TO TEST DATA SET
2604  007232  004737  021026                  JSR     PC,@#LDAFT      ; GO TEST
2605
2606  007236  000400                          BR      TST114          ;;
2607
2608                                  ; ( FOR DATA SEE TEST OF FP REG AC0 WITH TEST PATTERN LDAF-4, ABOVE )
2609
2610                                  ;;*****************************************************************
2611                                  ;*TEST 114      TEST FP AC3, F MODE, TEST PATTERN LDAF-5
2612                                  ;;*****************************************************************
2613  007240  000004                  TST114: SCOPE
2614  007242  012704  021274                  MOV     #LDARF3,R4      ; PTR TO TESTING ROUTINE
2615  007246  012705  006534                  MOV     #LDAF5,R5       ; PTR TO TEST DATA SET
2616  007252  004737  021026                  JSR     PC,@#LDAFT      ; GO TEST
2617
2618  007256  000400                          BR      TST115          ;;
2619
2620                                  ; ( FOR DATA SEE TEST OF FP REG AC0 WITH TEST PATTERN LDAF-5, ABOVE )
2621
2622                                  ;;*****************************************************************
2623                                  ;*TEST 115      TEST FP AC3, F MODE, TEST PATTERN LDAF-6
2624                                  ;;*****************************************************************
2625  007260  000004                  TST115: SCOPE
2626  007262  012704  021274                  MOV     #LDARF3,R4      ; PTR TO TESTING ROUTINE
2627  007266  012705  006606                  MOV     #LDAF6,R5       ; PTR TO TEST DATA SET
2628  007272  004737  021026                  JSR     PC,@#LDAFT      ; GO TEST
2629
2630  007276  000400                          BR      TST116          ;;
2631
2632                                  ; ( FOR DATA SEE TEST OF FP REG AC0 WITH TEST PATTERN LDAF-6, ABOVE )
2633
2634
2635                                  ;;*****************************************************************
2636                                  ;*TEST 116      TEST FP AC4, F MODE, TEST PATTERN LDAF-1
```

```
                                ;;*****************************************************************
007300  000004                  TST116: SCOPE
007302  012704  021316                  MOV     #LDARF4,R4      ; PTR TO TESTING ROUTINE
007306  012705  006264                  MOV     #LDAF1,R5       ; PTR TO TEST DATA SET
007312  004737  021026                  JSR     PC,@#LDAFT      ; GO TEST
007316  000400                          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
                                ;;*****************************************************************
007320  000004                  TST117: SCOPE
007322  012704  021316                  MOV     #LDARF4,R4      ; PTR TO TESTING ROUTINE
007326  012705  006336                  MOV     #LDAF2,R5       ; PTR TO TEST DATA SET
007332  004737  021026                  JSR     PC,@#LDAFT      ; GO TEST
007336  000400                          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
                                ;;*****************************************************************
007340  000004                  TST120: SCOPE
007342  012704  021316                  MOV     #LDARF4,R4      ; PTR TO TESTING ROUTINE
007346  012705  006410                  MOV     #LDAF3,R5       ; PTR TO TEST DATA SET
007352  004737  021026                  JSR     PC,@#LDAFT      ; GO TEST
007356  000400                          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
                                ;;*****************************************************************
007360  000004                  TST121: SCOPE
007362  012704  021316                  MOV     #LDARF4,R4      ; PTR TO TESTING ROUTINE
007366  012705  006462                  MOV     #LDAF4,R5       ; PTR TO TEST DATA SET
007372  004737  021026                  JSR     PC,@#LDAFT      ; GO TEST
007376  000400                          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
                                ;;*****************************************************************
007400  000004                  TST122: SCOPE
007402  012704  021316                  MOV     #LDARF4,R4      ; PTR TO TESTING ROUTINE
007406  012705  006534                  MOV     #LDAF5,R5       ; PTR TO TEST DATA SET
007412  004737  021026                  JSR     PC,@#LDAFT      ; GO TEST
007416  000400                          BR      TST123          ;;
```

```
                                        ; ( FOR DATA SEE TEST OF FP REG AC0 WITH TEST PATTERN LDAF-5, ABOVE )
                                        ;;*****************************************************************
                                        ;*TEST 123      TEST FP AC5, F MODE, TEST PATTERN LDAF-1
                                        ;;*****************************************************************
007420  000004                          TST123: SCOPE
007422  012704  021346                          MOV     #LDARF5,R4      ; PTR TO TESTING ROUTINE
007426  012705  006264                          MOV     #LDAF1,R5       ; PTR TO TEST DATA SET
007432  004737  021026                          JSR     PC,@#LDAFT      ; GO TEST
007436  000400                                  BR      TST124          ;;
                                        ; ( FOR DATA SEE TEST OF FP REG AC0 WITH TEST PATTERN LDAF-1, ABOVE )
                                        ;;*****************************************************************
                                        ;*TEST 124      TEST FP AC5, F MODE, TEST PATTERN LDAF-2
                                        ;;*****************************************************************
007440  000004                          TST124: SCOPE
007442  012704  021346                          MOV     #LDARF5,R4      ; PTR TO TESTING ROUTINE
007446  012705  006336                          MOV     #LDAF2,R5       ; PTR TO TEST DATA SET
007452  004737  021026                          JSR     PC,@#LDAFT      ; GO TEST
007456  000400                                  BR      TST125          ;;
                                        ; ( FOR DATA SEE TEST OF FP REG AC0 WITH TEST PATTERN LDAF-2, ABOVE )
                                        ;;*****************************************************************
                                        ;*TEST 125      TEST FP AC5, F MODE, TEST PATTERN LDAF-3
                                        ;;*****************************************************************
007460  000004                          TST125: SCOPE
007462  012704  021346                          MOV     #LDARF5,R4      ; PTR TO TESTING ROUTINE
007466  012705  006410                          MOV     #LDAF3,R5       ; PTR TO TEST DATA SET
007472  004737  021026                          JSR     PC,@#LDAFT      ; GO TEST
007476  000400                                  BR      TST126          ;;
                                        ; ( FOR DATA SEE TEST OF FP REG AC0 WITH TEST PATTERN LDAF-3, ABOVE )
                                        ;;*****************************************************************
                                        ;*TEST 126      TEST FP AC5, F MODE, TEST PATTERN LDAF-4
                                        ;;*****************************************************************
007500  000004                          TST126: SCOPE
007502  012704  021346                          MOV     #LDARF5,R4      ; PTR TO TESTING ROUTINE
007506  012705  006462                          MOV     #LDAF4,R5       ; PTR TO TEST DATA SET
007512  004737  021026                          JSR     PC,@#LDAFT      ; GO TEST
007516  000400                                  BR      TST127          ;;
                                        ; ( FOR DATA SEE TEST OF FP REG AC0 WITH TEST PATTERN LDAF-4, ABOVE )
                                        ;;*****************************************************************
                                        ;*TEST 127      TEST FP AC5, F MODE, TEST PATTERN LDAF-5
                                        ;;*****************************************************************
007520  000004                          TST127: SCOPE
007522  012704  021346                          MOV     #LDARF5,R4      ; PTR TO TESTING ROUTINE
```

```
                        T127    TEST FP AC5, F MODE, TEST PATTERN LDAF-5

2749  007526  012705  006534            MOV     #LDAF5,R5       ; PTR TO TEST DATA SET
2750  007532  004737  021026            JSR     PC,@#LDAFT      ; GO TEST
2751
2752  007536  000400                    BR      TST130          ;;
2753
2754                                  ; ( FOR DATA SEE TEST OF FP REG AC0 WITH TEST PATTERN LDAF-5, ABOVE )
2755
2756
2757
```

```
2758
2759                                    ;;*****************************************************************
2760                                            .DSABL  AMA             ; ASSEMBLE ALL REFERENCES AS THEY ARE PRINTED
2761                                    ;;*****************************************************************
2762
2763                                    ;;*****************************************************************
2764                                    ;*TEST 130       TEST OF LOAD-FSRC MODE-1, STORE-FDST MODE-6(PC), D MODE
2765                                    ;;*****************************************************************
2766  007540  000004                   TST130: SCOPE
2767  007542  170127  047600                   LDFPS   #047600         ; D MODE FPS
2768  007546  016767  000102  171414          MOV     ADRD1+0,$REG0   ; MOVE PATTERN
2769  007554  016767  000076  171410          MOV     ADRD1+2,$REG1   ;
2770  007562  016767  000072  171404          MOV     ADRD1+4,$REG2   ;
2771  007570  016767  000066  171400          MOV     ADRD1+6,$REG3   ;
2772
2773  007576  012705  001170                   MOV     #$REG0,R5       ; ADDR(DATA)
2774  007602  172415                           LDD     (R5),AC0        ; M1-R5
2775
2776  007604  174067  171370                   STD     AC0,$REG4       ; M6-R7
2777
2778  007610  026767  171354  171362          CMP     $REG0,$REG4     ; 1ST WORD PATTERN CHECK?
2779  007616  001014                           BNE     64$             ;
2780  007620  026767  171346  171354          CMP     $REG1,$REG5     ; 2ND WORD PATTERN CHECK?
2781  007626  001010                           BNE     64$             ;
2782  007630  026767  171340  171346          CMP     $REG2,$REG6     ; 3RD WORD PATTERN CHECK?
2783  007636  001004                           BNE     64$             ;
2784  007640  026767  171332  171340          CMP     $REG3,$REG7     ; 4TH WORD PATTERN CHECK?
2785  007646  001401                           BEQ     65$             ;
2786  007650  104031                   64$:    ERROR   31              ; PATTERN DOESNT MATCH
2787
2788  007652                           65$:
2789  007652  000404                           BR      TST131          ;;
2790
2791  007654  052525  052525  052525  ADRD1:  .WORD   ALTP,ALTP,ALTP,ALTP    ; TEST PATTERN
2792  007662  052525
2793
2794
2795                                    ;;*****************************************************************
2796                                    ;*TEST 131       TEST OF LOAD-FSRC MODE-4, STORE-FDST MODE-3(PC), D MODE
2797                                    ;;*****************************************************************
2798  007664  000004                   TST131: SCOPE
2799  007666  170127  047600                   LDFPS   #047600         ; D MODE FPS
2800  007672  016767  000112  171270          MOV     ADRD2+0,$REG0   ; MOVE PATTERN
2801  007700  016767  000106  171264          MOV     ADRD2+2,$REG1   ;
2802  007706  016767  000102  171260          MOV     ADRD2+4,$REG2   ;
2803  007714  016767  000076  171254          MOV     ADRD2+6,$REG3   ;
2804
2805  007722  012704  001200                   MOV     #$REG0+10,R4    ; ADDR(DATA+10)
2806  007726  172544                           LDD     -(R4),AC1       ; M4-R4
2807
2808  007730  174137  001200                   STD     AC1,@#$REG4     ; M3-R7
2809
2810  007734  020427  001170                   CMP     R4,#$REG0       ; WAS FSRC ADDR REG DECRE RIGHT AMOUNT?
2811  007740  001401                           BEQ     66$             ;
2812  007742  104032                           ERROR   32              ; NOT EQUAL, SIGNAL ERROR
2813  007744                           66$:
```

```
FPU BASIC INSTR TESTS   MACY11 27(1006)  09-FEB-77  09:48  PAGE 61
DQFPAA.P11    09-FEB-77 09:42            T131    TEST OF LOAD-FSRC MODE-4, STORE-FDST MODE-3(PC), D MODE

 2814
 2815  007744  026767  171220  171226          CMP     $REG0,$REG4     ; 1ST WORD PATTERN CHECK?
 2816  007752  001014                          BNE     64$             ;
 2817  007754  026767  171212  171220          CMP     $REG1,$REG5     ; 2ND WORD PATTERN CHECK?
 2818  007762  001010                          BNE     64$             ;
 2819  007764  026767  171204  171212          CMP     $REG2,$REG6     ; 3RD WORD PATTERN CHECK?
 2820  007772  001004                          BNE     64$             ;
 2821  007774  026767  171176  171204          CMP     $REG3,$REG7     ; 4TH WORD PATTERN CHECK?
 2822  010002  001401                          BEQ     65$             ;
 2823  010004  104033                  64$:    ERROR   33              ; PATTERN DOESNT MATCH
 2824
 2825  010006                           65$:
 2826  010006  000404                          BR      TST132          ;;
 2827
 2828  010010  170360  170360  170360  ADRD2:  .WORD   ALT4N,ALT4N,ALT4N,ALT4N ; TEST PATTERN
 2829  010016  170360
 2830
 2831
 2832                                   ;;**************************************************************
 2833                                   ;*TEST 132      TEST OF LOAD-FSRC MODE-7, STORE-FDST MODE-3, D MODE
 2834                                   ;;**************************************************************
 2835  010020  000004                  TST132: SCOPE
 2836  010022  170127  047600                  LDFPS   #047600         ; D MODE FPS
 2837  010026  016767  000132  171134          MOV     ADRD3+0,$REG0   ; MOVE PATTERN
 2838  010034  016767  000126  171130          MOV     ADRD3+2,$REG1   ;
 2839  010042  016767  000122  171124          MOV     ADRD3+4,$REG2   ;
 2840  010050  016767  000116  171120          MOV     ADRD3+6,$REG3   ;
 2841
 2842  010056  012767  001170  171124          MOV     #$REG0,$REG10   ; ADDR(DATA)
 2843  010064  012705  001214                  MOV     #$REG10+4,R5    ; ADDR(ADDR(DATA))+4
 2844  010070  172675  177774                  LDD     @-4(R5),AC2     ; M7-R5
 2845
 2846  010074  012767  001200  171110          MOV     #$REG4,$REG11   ; ADDR(DEST)
 2847  010102  012704  001212                  MOV     #$REG11,R4      ; ADDR(ADDR(DEST))
 2848  010106  174234                          STD     AC2,@(R4)+      ; M3-R4
 2849
 2850  010110  020427  001214                  CMP     R4,#$REG11+2    ; WAS FDST ADDR REG INCRE RIGHT AMOUNT?
 2851  010114  001401                          BEQ     66$             ;
 2852  010116  104034                          ERROR   34              ; NOT EQUAL, SIGNAL ERROR
 2853  010120                           66$:
 2854
 2855  010120  026767  171044  171052          CMP     $REG0,$REG4     ; 1ST WORD PATTERN CHECK?
 2856  010126  001014                          BNE     64$             ;
 2857  010130  026767  171036  171044          CMP     $REG1,$REG5     ; 2ND WORD PATTERN CHECK?
 2858  010136  001010                          BNE     64$             ;
 2859  010140  026767  171030  171036          CMP     $REG2,$REG6     ; 3RD WORD PATTERN CHECK?
 2860  010146  001004                          BNE     64$             ;
 2861  010150  026767  171022  171030          CMP     $REG3,$REG7     ; 4TH WORD PATTERN CHECK?
 2862  010156  001401                          BEQ     65$             ;
 2863  010160  104035                  64$:    ERROR   35              ; PATTERN DOESNT MATCH
 2864
 2865  010162                           65$:
 2866  010162  000404                          BR      TST133          ;;
 2867
 2868  010164  125252  125252  125252  ADRD3:  .WORD   ALTN,ALTN,ALTN,ALTN     ; TEST PATTERN
 2869  010172  125252
```

```
2870
2871
2872                                    ;;*****************************************************************
2873                                    ;*TEST 133      TEST OF LOAD-FSRC MODE-6(PC), STORE-FDST MODE-2, D MODE
2874                                    ;*****************************************************************
2875  010174  000004                   TST133: SCOPE
2876  010176  170127  047600                   LDFPS   #047600          ; D MODE FPS
2877  010202  016767  000112  170760           MOV     ADRD4+0,$REG0    ; MOVE PATTERN
2878  010210  016767  000106  170754           MOV     ADRD4+2,$REG1    ;
2879  010216  016767  000102  170750           MOV     ADRD4+4,$REG2    ;
2880  010224  016767  000076  170744           MOV     ADRD4+6,$REG3    ;
2881
2882  010232  172767  170732                   LDD     $REG0,AC3        ; M6-R7
2883
2884  010236  012705  001200                   MOV     #$REG4,R5        ; ADDR(DEST)
2885  010242  174325                           STD     AC3,(R5)+        ; M2-R5
2886
2887  010244  020527  001210                   CMP     R5,#$REG4+10     ; WAS FDST ADDR REG INCRE RIGHT AMOUNT?
2888  010250  001401                           BEQ     66$              ;
2889  010252  104036                           ERROR   36               ; NOT EQUAL, SIGNAL ERROR
2890  010254                           66$:
2891
2892  010254  026767  170710  170716           CMP     $REG0,$REG4      ; 1ST WORD PATTERN CHECK?
2893  010262  001014                           BNE     64$              ;
2894  010264  026767  170702  170710           CMP     $REG1,$REG5      ; 2ND WORD PATTERN CHECK?
2895  010272  001010                           BNE     64$              ;
2896  010274  026767  170674  170702           CMP     $REG2,$REG6      ; 3RD WORD PATTERN CHECK?
2897  010302  001004                           BNE     64$              ;
2898  010304  026767  170666  170674           CMP     $REG3,$REG7      ; 4TH WORD PATTERN CHECK?
2899  010312  001401                           BEQ     65$              ;
2900  010314  104031                   64$:    ERROR   31               ; PATTERN DOESNT MATCH
2901
2902  010316                           65$:
2903  010316  000404                           BR      TST134           ;;
2904
2905  010320  007417  007417  007417   ADRD4:  .WORD   ALT4P,ALT4P,ALT4P,ALT4P ; TEST PATTERN
2906  010326  007417
2907
2908
2909                                    ;;*****************************************************************
2910                                    ;*TEST 134      TEST OF LOAD-FSRC MODE-2, STORE-FDST MODE-1, D MODE
2911                                    ;*****************************************************************
2912  010330  000004                   TST134: SCOPE
2913  010332  170127  047600                   LDFPS   #047600          ; D MODE FPS
2914  010336  016767  000114  170624           MOV     ADRD5+0,$REG0    ; MOVE PATTERN
2915  010344  016767  000110  170620           MOV     ADRD5+2,$REG1    ;
2916  010352  016767  000104  170614           MOV     ADRD5+4,$REG2    ;
2917  010360  016767  000100  170610           MOV     ADRD5+6,$REG3    ;
2918
2919  010366  012700  001170                   MOV     #$REG0,R0        ; ADDR(DATA)
2920  010372  172420                           LDD     (R0)+,AC0        ; M2-R0
2921
2922  010374  012703  001200                   MOV     #$REG4,R3        ; ADDR(DEST)
2923  010400  174013                           STD     AC0,(R3)         ; M1-R3
2924
2925  010402  020027  001200                   CMP     R0,#$REG0+10     ; WAS FSRC ADDR REG INCRE RIGHT AMOUNT?
```

```
FPU BASIC INSTR TESTS   MACY11 27(1006)  09-FEB-77  09:48  PAGE 63
DQFPAA.P11    09-FEB-77 09:42            T134    TEST OF LOAD-FSRC MODE-2, STORE-FDST MODE-1, D MODE

  2926  010406  001401                          BEQ     66$             ;
  2927  010410  104037                          ERROR   37              ; NOT EQUAL, SIGNAL ERROR
  2928  010412                          66$:
  2929
  2930  010412  026767  170552  170560          CMP     $REG0,$REG4     ; 1ST WORD PATTERN CHECK?
  2931  010420  001014                          BNE     64$             ;
  2932  010422  026767  170544  170552          CMP     $REG1,$REG5     ; 2ND WORD PATTERN CHECK?
  2933  010430  001010                          BNE     64$             ;
  2934  010432  026767  170536  170544          CMP     $REG2,$REG6     ; 3RD WORD PATTERN CHECK?
  2935  010440  001004                          BNE     64$             ;
  2936  010442  026767  170530  170536          CMP     $REG3,$REG7     ; 4TH WORD PATTERN CHECK?
  2937  010450  001401                          BEQ     65$             ;
  2938  010452  104040                  64$:    ERROR   40              ; PATTERN DOESNT MATCH
  2939
  2940  010454                          65$:
  2941  010454  000404                          BR      TST135          ;;
  2942
  2943  010456  125252  125252  125252  ADRD5:  .WORD   ALTN,ALTN,ALTN,ALTN     ; TEST PATTERN
  2944  010464  125252
  2945
  2946
  2947                                  ;;**************************************************************
  2948                                  ;*TEST 135       TEST OF LOAD-FSRC MODE-5, STORE-FDST MODE-7(PC), D MODE
  2949                                  ;;**************************************************************
  2950  010466  000004                  TST135: SCOPE
  2951  010470  170127  047600                  LDFPS   #047600         ; D MODE FPS
  2952  010474  016767  000126  170466          MOV     ADRD6+0,$REG0   ; MOVE PATTERN
  2953  010502  016767  000122  170462          MOV     ADRD6+2,$REG1   ;
  2954  010510  016767  000116  170456          MOV     ADRD6+4,$REG2   ;
  2955  010516  016767  000112  170452          MOV     ADRD6+6,$REG3   ;
  2956
  2957  010524  012767  001170  170456          MOV     #$REG0,$REG10   ; ADDR(DATA)
  2958  010532  012701  001212                  MOV     #$REG10+2,R1    ; ADDR(ADDR(DATA)+2)
  2959  010536  172551                          LDD     @-(R1),AC1      ; M5-R1
  2960
  2961  010540  012767  001200  170444          MOV     #$REG4,$REG11   ; ADDR(DEST)
  2962  010546  174177  170440                  STD     AC1,@$REG11     ; M7-R7
  2963
  2964  010552  020127  001210                  CMP     R1,#$REG10      ; WAS FSRC ADDR REG DECRE RIGHT AMOUNT?
  2965  010556  001401                          BEQ     66$             ;
  2966  010560  104041                          ERROR   41              ; NOT EQUAL, SIGNAL ERROR
  2967  010562                          66$:
  2968
  2969  010562  026767  170402  170410          CMP     $REG0,$REG4     ; 1ST WORD PATTERN CHECK?
  2970  010570  001014                          BNE     64$             ;
  2971  010572  026767  170374  170402          CMP     $REG1,$REG5     ; 2ND WORD PATTERN CHECK?
  2972  010600  001010                          BNE     64$             ;
  2973  010602  026767  170366  170374          CMP     $REG2,$REG6     ; 3RD WORD PATTERN CHECK?
  2974  010610  001004                          BNE     64$             ;
  2975  010612  026767  170360  170366          CMP     $REG3,$REG7     ; 4TH WORD PATTERN CHECK?
  2976  010620  001401                          BEQ     65$             ;
  2977  010622  104042                  64$:    ERROR   42              ; PATTERN DOESNT MATCH
  2978
  2979  010624                          65$:
  2980  010624  000404                          BR      TST136          ;;
  2981
```

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

  2982  010623  007417  007417  007417  ADRD6:  .WORD   ALT4P,ALT4P,ALT4P,ALT4P ; TEST PATTERN
  2983  010634  007417
  2984
  2985
  2986                                  ;;*****************************************************************
  2987                                  ;*TEST 136      TEST OF LOAD-FSRC MODE-2(PC), STORE-FDST MODE-7, D MODE
  2988                                  ;;*****************************************************************
  2989  010636  000004                  TST136: SCOPE
  2990  010640  170127  047600                  LDFPS   #047600         ; D MODE FPS
  2991  010644  016767  000130  170316          MOV     ADRD7+0,$REG0   ; MOVE PATTERN
  2992  010652  016767  000124  170312          MOV     ADRD7+2,$REG1   ;
  2993  010660  016767  000120  170306          MOV     ADRD7+4,$REG2   ;
  2994  010666  016767  000114  170302          MOV     ADRD7+6,$REG3   ;
  2995
  2996  010674  016767  170270  000002          MOV     $REG0,1$        ; PUT DATA
  2997  010702  172627                          LDD     (PC)+,AC2       ; M2-R7
  2998  010704  000000                  1$:     .WORD   0               ; DATA
  2999  010706  000403                          BR      2$              ; THIS SHOULD BE NEXT INSTR AFTER FPP PC MODE 2
  3000  010710  104120                          ERROR   120             ; NOT HERE  (+4 INCRE)
  3001  010712  104121                          ERROR   121             ; OR HERE  (+6 INCRE)
  3002  010714  104122                          ERROR   122             ; OR HERE  (+10 INCRE)
  3003
  3004  010716  012767  001200  170264  2$:     MOV     #$REG4,$REG10   ; ADDR(DEST)
  3005  010724  012700  001152                  MOV     #$REG10-36,R0   ; ADDR(ADDR(DEST))-36
  3006  010730  174270  000036                  STD     AC2,@36(R0)     ; M7-R0
  3007
  3008  010734  026767  170230  170236          CMP     $REG0,$REG4     ; 1ST WORD PATTERN CHECK?
  3009  010742  001014                          BNE     64$             ;
  3010  010744  026767  170222  170230          CMP     $REG1,$REG5     ; 2ND WORD PATTERN CHECK?
  3011  010752  001010                          BNE     64$             ;
  3012  010754  026767  170214  170222          CMP     $REG2,$REG6     ; 3RD WORD PATTERN CHECK?
  3013  010762  001004                          BNE     64$             ;
  3014  010764  026767  170206  170214          CMP     $REG3,$REG7     ; 4TH WORD PATTERN CHECK?
  3015  010772  001401                          BEQ     65$             ;
  3016  010774  104043                  64$:    ERROR   43              ; PATTERN DOESNT MATCH
  3017
  3018  010776                          65$:
  3019  010776  000404                          BR      TST137          ;;
  3020
  3021  011000  104117  000000  000000  ADRD7:  .WORD   ERROR117,0,0,0  ; TEST PATTERN
  3022  011006  000000
  3023
  3024
  3025                                  ;;*****************************************************************
  3026                                  ;*TEST 137      TEST OF LOAD-FSRC MODE-7(PC), STORE-FDST MODE-6, D MODE
  3027                                  ;;*****************************************************************
  3028  011010  000004                  TST137: SCOPE
  3029  011012  170127  047600                  LDFPS   #047600         ; D MODE FPS
  3030  011016  016767  000112  170144          MOV     ADRD10+0,$REG0  ; MOVE PATTERN
  3031  011024  016767  000106  170140          MOV     ADRD10+2,$REG1  ;
  3032  011032  016767  000102  170134          MOV     ADRD10+4,$REG2  ;
  3033  011040  016767  000076  170130          MOV     ADRD10+6,$REG3  ;
  3034
  3035  011046  012767  001170  170134          MOV     #$REG0,$REG10   ; ADDR(DATA)
  3036  011054  172777  170130                  LDD     @$REG10,AC3     ; M7-R7
  3037
```

```
3038  011060  012704  001212              MOV     #$REG4+12,R4    ; ADDR(DEST)+12
3039  011064  174364  177766              STD     AC3,-12(R4)     ; M6-R4
3040
3041  011070  026767  170074  170102      CMP     $REG0,$REG4     ; 1ST WORD PATTERN CHECK?
3042  011076  001014                      BNE     64$             ;
3043  011100  026767  170066  170074      CMP     $REG1,$REG5     ; 2ND WORD PATTERN CHECK?
3044  011106  001010                      BNE     64$             ;
3045  011110  026767  170060  170066      CMP     $REG2,$REG6     ; 3RD WORD PATTERN CHECK?
3046  011116  001004                      BNE     64$             ;
3047  011120  026767  170052  170060      CMP     $REG3,$REG7     ; 4TH WORD PATTERN CHECK?
3048  011126  001401                      BEQ     65$             ;
3049  011130  104044            64$:      ERROR   44              ; PATTERN DOESNT MATCH
3050
3051  011132                    65$:
3052  011132  000404                      BR      TST140          ;;
3053
3054  011134  170360  170360  170360  ADRD10: .WORD   ALT4N,ALT4N,ALT4N,ALT4N ; TEST PATTERN
3055  011142  170360
3056
3057
3058                            ;;*****************************************************************
3059                            ;*TEST 140      TEST OF LOAD-FSRC MODE-3, STORE-FDST MODE-2(PC), D MODE
3060                            ;;*****************************************************************
3061  011144  000004            TST140: SCOPE
3062  011146  170127  047600              LDFPS   #047600         ; D MODE FPS
3063  011152  016767  000114  170010      MOV     ADRD11+0,$REG0  ; MOVE PATTERN
3064  011160  016767  000110  170004      MOV     ADRD11+2,$REG1  ;
3065  011166  016767  000104  170000      MOV     ADRD11+4,$REG2  ;
3066  011174  016767  000100  167774      MOV     ADRD11+6,$REG3  ;
3067
3068  011202  012767  001170  170000      MOV     #$REG0,$REG10   ; ADDR(DATA)
3069  011210  012702  001210              MOV     #$REG10,R2      ; ADDR(ADDR(DATA))
3070  011214  172432                      LDD     @(R2)+,AC0      ; M3-R2
3071
3072  011216  012767  104117  000002      MOV     #ERROR117,1$    ; SETUP ERROR CALL FOR WRONG INCREMENT
3073  011224  174027                      STD     AC0,(PC)+       ; M2-R7
3074  011226  000000            1$:       .WORD   0               ; DEST
3075  011230  000403                      BR      2$              ; THIS SHOULD BE NEXT INSTR AFTER FPP PC MODE 2
3076  011232  104120                      ERROR   120             ; NOT HERE  (+4 INCRE)
3077  011234  104121                      ERROR   121             ; OR HERE  (+6 INCRE)
3078  011236  104122                      ERROR   122             ; OR HERE  (+10 INCRE)
3079  011240  016767  177762  167732  2$: MOV     1$,$REG4        ; STORE DEST
3080
3081  011246  020227  001212              CMP     R2,#$REG10+2    ; WAS FSRC ADDR REG INCRE BY RIGHT AMOUNT?
3082  011252  001401                      BEQ     64$             ;
3083  011254  104045                      ERROR   45              ; NOT EQUAL, SIGNAL ERROR
3084  011256                    64$:
3085
3086  011256  026767  167706  167714      CMP     $REG0,$REG4     ; WAS 1 WORD STORED OK?
3087  011264  001401                      BEQ     65$             ;
3088  011266  104046                      ERROR   46              ; NOT EQUAL, SIGNAL ERROR
3089  011270                    65$:
3090
3091  011270  000404                      BR      TST141          ;;
3092
3093  011272  052525  000000  000000  ADRD11: .WORD   ALTP,0,0,0      ; TEST PATTERN
```

```
3094  011300  000000
3095
3096
3097                                    ;;*****************************************************************
3098                                    ;*TEST 141      TEST OF LOAD-FSRC MODE-6, STORE-FDST MODE-4, D MODE
3099                                    ;;*****************************************************************
3100  011302  000004                    TST141: SCOPE
3101  011304  170127  047600                    LDFPS   #047600          ; D MODE FPS
3102  011310  016767  000116  167652           MOV     ADRD12+0,$REG0   ; MOVE PATTERN
3103  011316  016767  000112  167646           MOV     ADRD12+2,$REG1   ;
3104  011324  016767  000106  167642           MOV     ADRD12+4,$REG2   ;
3105  011332  016767  000102  167636           MOV     ADRD12+6,$REG3   ;
3106
3107  011340  012703  001262                    MOV     #$REG0+72,R3     ; ADDR(DATA)+72
3108  011344  172563  177706                    LDD     -72(R3),AC1      ; M6-R3
3109
3110  011350  012701  001210                    MOV     #$REG4+10,R1     ; ADDR(DEST+10)
3111  011354  174141                            STD     AC1,-(R1)        ; M4-R1
3112
3113  011356  020127  001200                    CMP     R1,#$REG4        ; WAS FDST ADDR REG DECRE RIGHT AMOUNT?
3114  011362  001401                            BEQ     66$              ;
3115  011364  104047                            ERROR   47               ; NOT EQUAL, SIGNAL ERROR
3116  011366                          66$:
3117
3118  011366  026767  167576  167604           CMP     $REG0,$REG4      ; 1ST WORD PATTERN CHECK?
3119  011374  001014                            BNE     64$              ;
3120  011376  026767  167570  167576           CMP     $REG1,$REG5      ; 2ND WORD PATTERN CHECK?
3121  011404  001010                            BNE     64$              ;
3122  011406  026767  167562  167570           CMP     $REG2,$REG6      ; 3RD WORD PATTERN CHECK?
3123  011414  001004                            BNE     64$              ;
3124  011416  026767  167554  167562           CMP     $REG3,$REG7      ; 4TH WORD PATTERN CHECK?
3125  011424  001401                            BEQ     65$              ;
3126  011426  104050                  64$:      ERROR   50               ; PATTERN DOESNT MATCH
3127
3128  011430                          65$:
3129  011430  000404                            BR      TST142           ;;
3130
3131  011432  170360  170360  170360  ADRD12: .WORD   ALT4N,ALT4N,ALT4N,ALT4N ; TEST PATTERN
3132  011440  170360
3133
3134
3135                                    ;;*****************************************************************
3136                                    ;*TEST 142      TEST OF LOAD-FSRC MODE-3(PC), STORE-FDST MODE-0, D MODE
3137                                    ;;*****************************************************************
3138  011442  000004                    TST142: SCOPE
3139  011444  170127  047600                    LDFPS   #047600          ; D MODE FPS
3140  011450  016767  000102  167512           MOV     ADRD13+0,$REG0   ; MOVE PATTERN
3141  011456  016767  000076  167506           MOV     ADRD13+2,$REG1   ;
3142  011464  016767  000072  167502           MOV     ADRD13+4,$REG2   ;
3143  011472  016767  000066  167476           MOV     ADRD13+6,$REG3   ;
3144
3145  011500  172637  001170                    LDD     @#$REG0,AC2      ; M3-R7
3146
3147  011504  174203                            STD     AC2,AC3          ; M0-R3
3148  011506  174367  167466                    STD     AC3,$REG4        ; M6-R7 WORKS
3149
```

```
3150  011512  026767  167452  167460          CMP     $REG0,$REG4     ; 1ST WORD PATTERN CHECK?
3151  011520  001014                          BNE     64$             ;
3152  011522  026767  167444  167452          CMP     $REG1,$REG5     ; 2ND WORD PATTERN CHECK?
3153  011530  001010                          BNE     64$             ;
3154  011532  026767  167436  167444          CMP     $REG2,$REG6     ; 3RD WORD PATTERN CHECK?
3155  011540  001004                          BNE     64$             ;
3156  011542  026767  167430  167436          CMP     $REG3,$REG7     ; 4TH WORD PATTERN CHECK?
3157  011550  001401                          BEQ     65$             ;
3158  011552  104051                  64$:    ERROR   51              ; PATTERN DOESNT MATCH
3159
3160  011554                           65$:
3161  011554  000404                          BR      TST143          ;;
3162
3163  011556  125252  125252  125252  ADRD13: .WORD   ALTN,ALTN,ALTN,ALTN     ; TEST PATTERN
3164  011564  125252
3165
3166
3167                                   ;;*****************************************************************
3168                                   ;*TEST 143      TEST OF LOAD-FSRC MODE-0, STORE-FDST MODE-5, D MODE
3169                                   ;;*****************************************************************
3170  011566  000004                   TST143: SCOPE
3171  011570  170127  047600                  LDFPS   #047600         ; D MODE FPS
3172  011574  016767  000122  167366          MOV     ADRD14+0,$REG0  ; MOVE PATTERN
3173  011602  016767  000116  167362          MOV     ADRD14+2,$REG1  ;
3174  011610  016767  000112  167356          MOV     ADRD14+4,$REG2  ;
3175  011616  016767  000106  167352          MOV     ADRD14+6,$REG3  ;
3176
3177  011624  172467  167340                  LDD     $REG0,AC0       ; M6-R7 WORKS
3178  011630  172500                          LDD     AC0,AC1         ; M0-R0
3179
3180  011632  012767  001200  167350          MOV     #$REG4,$REG10   ; ADDR(DEST)
3181  011640  012702  001212                  MOV     #$REG10+2,R2    ; ADDR(ADDR(DEST)+2)
3182  011644  174152                          STD     AC1,@-(R2)      ; M5-R2
3183
3184  011646  020227  001210                  CMP     R2,#$REG10      ; WAS FDST ADDR REG DECRE RIGHT AMOUNT?
3185  011652  001401                          BEQ     66$             ;
3186  011654  104052                          ERROR   52              ; NOT EQUAL, SIGNAL ERROR
3187  011656                           66$:
3188
3189  011656  026767  167306  167314          CMP     $REG0,$REG4     ; 1ST WORD PATTERN CHECK?
3190  011664  001014                          BNE     64$             ;
3191  011666  026767  167300  167306          CMP     $REG1,$REG5     ; 2ND WORD PATTERN CHECK?
3192  011674  001010                          BNE     64$             ;
3193  011676  026767  167272  167300          CMP     $REG2,$REG6     ; 3RD WORD PATTERN CHECK?
3194  011704  001004                          BNE     64$             ;
3195  011706  026767  167264  167272          CMP     $REG3,$REG7     ; 4TH WORD PATTERN CHECK?
3196  011714  001401                          BEQ     65$             ;
3197  011716  104046                  64$:    ERROR   46              ; PATTERN DOESNT MATCH
3198
3199  011720                           65$:
3200  011720  000404                          BR      TST144          ;;
3201
3202  011722  125252  125252  125252  ADRD14: .WORD   ALTN,ALTN,ALTN,ALTN     ; TEST PATTERN
3203  011730  125252
3204
3205
```

```
                                        ;;*****************************************************************
                                        ;*TEST 144      TEST OF LOAD-FSRC MODE-1, STORE-FDST MODE-4, F MODE
                                        ;;*****************************************************************
011732  000004                  TST144: SCOPE
011734  170127  047400                  LDFPS   #047400         ; F MODE FPS
011740  016767  000060  167222          MOV     ADRF1+0,$REG0   ; MOVE PATTERN
011746  016767  000054  167216          MOV     ADRF1+2,$REG1   ;
                                        ;
011754  012702  001170                  MOV     #$REG0,R2       ; ADDR(DATA)
011760  172412                          LDF     (R2),AC0        ; M1-R2
                                        ;
011762  012705  001200                  MOV     #$REG2+4,R5     ; ADDR(DEST+4)
011766  174045                          STF     AC0,-(R5)       ; M4-R5
                                        ;
011770  020527  001174                  CMP     R5,#$REG2       ; WAS FDST ADDR REG DECRE RIGHT AMOUNT?
011774  001401                          BEQ     66$             ;
011776  104053                          ERROR   53              ; NOT EQUAL, SIGNAL ERROR
012000                          66$:
                                        ;
012000  026767  167164  167166          CMP     $REG0,$REG2     ; 1ST WORD PATTERN CHECK?
012006  001004                          BNE     64$             ;
012010  026767  167156  167160          CMP     $REG1,$REG3     ; 2ND WORD PATTERN CHECK?
012016  001401                          BEQ     65$             ;
012020  104054                  64$:    ERROR   54              ; PATTERN DOESNT MATCH
                                        ;
012022                          65$:
012022  000402                          BR      TST145          ;;
                                        ;
012024  170360  170360          ADRF1:  .WORD   ALT4N,ALT4N     ; TEST PATTERN
                                        ;
                                        ;
                                        ;;*****************************************************************
                                        ;*TEST 145      TEST OF LOAD-FSRC MODE-3, STORE-FDST MODE-6, F MODE
                                        ;;*****************************************************************
012030  000004                  TST145: SCOPE
012032  170127  047400                  LDFPS   #047400         ; F MODE FPS
012036  016767  000070  167124          MOV     ADRF2+0,$REG0   ; MOVE PATTERN
012044  016767  000064  167120          MOV     ADRF2+2,$REG1   ;
                                        ;
012052  012767  001170  167120          MOV     #$REG0,$REG4    ; ADDR(DATA)
012060  012704  001200                  MOV     #$REG4,R4       ; ADDR(ADDR(DATA))
012064  172534                          LDF     @(R4)+,AC1      ; M3-R4
                                        ;
012066  012700  001166                  MOV     #$REG2-6,R0     ; ADDR(DEST)-6
012072  174160  000006                  STF     AC1,6(R0)       ; M6-R0
                                        ;
012076  020427  001202                  CMP     R4,#$REG4+2     ; WAS FSRC ADDR REG INCRE RIGHT AMOUNT?
012102  001401                          BEQ     66$             ;
012104  104055                          ERROR   55              ; NOT EQUAL, SIGNAL ERROR
012106                          66$:
                                        ;
012106  026767  167056  167060          CMP     $REG0,$REG2     ; 1ST WORD PATTERN CHECK?
012114  001004                          BNE     64$             ;
012116  026767  167050  167052          CMP     $REG1,$REG3     ; 2ND WORD PATTERN CHECK?
012124  001401                          BEQ     65$             ;
012126  104056                  64$:    ERROR   56              ; PATTERN DOESNT MATCH
```

```
3262
3263  012130                          65$:
3264  012130  000402                          BR      TST146          ;;
3265
3266  012132  052525  052525          ADRF2:  .WORD   ALTP,ALTP       ; TEST PATTERN
3267
3268
3269                                  ;;*****************************************************************
3270                                  ;*TEST 146       TEST OF LOAD-FSRC MODE-5, STORE-FDST MODE-3, F MODE
3271                                  ;;*****************************************************************
3272  012136  000004                  TST146: SCOPE
3273  012140  170127  047400                  LDFPS   #047400         ; F MODE FPS
3274  012144  016767  000104  167016          MOV     ADRF3+0,$REG0   ; MOVE PATTERN
3275  012152  016767  000100  167012          MOV     ADRF3+2,$REG1   ;
3276
3277  012160  012767  001170  167012          MOV     #$REG0,$REG4    ; ADDR(DATA)
3278  012166  012700  001202                  MOV     #$REG4+2,R0     ; ADDR(ADDR(DATA)+2)
3279  012172  172650                          LDF     @-(R0),AC2      ; M5-R0
3280
3281  012174  012767  001174  167000          MOV     #$REG2,$REG5    ; ADDR(DEST)
3282  012202  012702  001202                  MOV     #$REG5,R2       ; ADDR(ADDR(DEST))
3283  012206  174232                          STF     AC2,@(R2)+      ; M3-R2
3284
3285  012210  020027  001200                  CMP     R0,#$REG4       ; WAS FSRC ADDR REG DECRE RIGHT AMOUNT?
3286  012214  001401                          BEQ     66$             ;
3287  012216  104057                          ERROR   57              ; NOT EQUAL, SIGNAL ERROR
3288  012220                          66$:
3289
3290  012220  020227  001204                  CMP     R2,#$REG5+2     ; WAS FDST ADDR REG INCRE RIGHT AMOUNT?
3291  012224  001401                          BEQ     67$             ;
3292  012226  104060                          ERROR   60              ; NOT EQUAL, SIGNAL ERROR
3293  012230                          67$:
3294
3295  012230  026767  166734  166736          CMP     $REG0,$REG2     ; 1ST WORD PATTERN CHECK?
3296  012236  001004                          BNE     64$             ;
3297  012240  026767  166726  166730          CMP     $REG1,$REG3     ; 2ND WORD PATTERN CHECK?
3298  012246  001401                          BEQ     65$             ;
3299  012250  104061                  64$:    ERROR   61              ; PATTERN DOESNT MATCH
3300
3301  012252                          65$:
3302  012252  000402                          BR      TST147          ;;
3303
3304  012254  007417  007417          ADRF3:  .WORD   ALT4P,ALT4P     ; TEST PATTERN
3305
3306
3307                                  ;;*****************************************************************
3308                                  ;*TEST 147       TEST OF LOAD-FSRC MODE-7, STORE-FDST MODE-1, F MODE
3309                                  ;;*****************************************************************
3310  012260  000004                  TST147: SCOPE
3311  012262  170127  047400                  LDFPS   #047400         ; F MODE FPS
3312  012266  016767  000060  166674          MOV     ADRF4+0,$REG0   ; MOVE PATTERN
3313  012274  016767  000054  166670          MOV     ADRF4+2,$REG1   ;
3314
3315  012302  012767  001170  166670          MOV     #$REG0,$REG4    ; ADDR(DATA)
3316  012310  012702  001162                  MOV     #$REG4-16,R2    ; ADDR(ADDR(DATA))-16
3317  012314  172772  000016                  LDF     @16(R2),AC3     ; M7-R2
```

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

```
3374
3375  012506  012767  001174  166466          MOV     #$REG2,$REG5    ; ADDR(DEST)
3376  012514  012705  001226                  MOV     #$REG5+24,R5    ; ADDR(ADDR(DEST))+24
3377  012520  174175  177754                  STF     AC1,@-24(R5)    ; M7-R5
3378
3379  012524  026767  166440  166442          CMP     $REG0,$REG2     ; 1ST WORD PATTERN CHECK?
3380  012532  001004                          BNE     64$             ;
3381  012534  026767  166432  166434          CMP     $REG1,$REG3     ; 2ND WORD PATTERN CHECK?
3382  012542  001401                          BEQ     65$             ;
3383  012544  104065                  64$:    ERROR   65              ; PATTERN DOESNT MATCH
3384
3385  012546                          65$:
3386  012546  000402                          BR      TST152          ;;
3387
3388  012550  125252  125252          ADRF6:  .WORD   ALTN,ALTN       ; TEST PATTERN
3389
3390
3391                                  ;;**************************************************************
3392                                  ;*TEST 152      TEST OF LOAD-FSRC MODE-2, STORE-FDST MODE-3(PC), F MODE
3393                                  ;;**************************************************************
3394  012554  000004                  TST152: SCOPE
3395  012556  170127  047400                  LDFPS   #047400         ; F MODE FPS
3396  012562  016767  000056  166400          MOV     ADRF7+0,$REG0   ; MOVE PATTERN
3397  012570  016767  000052  166374          MOV     ADRF7+2,$REG1   ;
3398
3399  012576  012703  001170                  MOV     #$REG0,R3       ; ADDR(DATA)
3400  012602  172623                          LDF     (R3)+,AC2       ; M2-R3
3401
3402  012604  174237  001174                  STF     AC2,@#$REG2     ; M3-R7
3403
3404  012610  020327  001174                  CMP     R3,#$REG0+4     ; WAS FSRC ADDR REG INCRE RIGHT AMOUNT?
3405  012614  001401                          BEQ     66$             ;
3406  012616  104066                          ERROR   66              ; NOT EQUAL, SIGNAL ERROR
3407  012620                          66$:
3408
3409  012620  026767  166344  166346          CMP     $REG0,$REG2     ; 1ST WORD PATTERN CHECK?
3410  012626  001004                          BNE     64$             ;
3411  012630  026767  166336  166340          CMP     $REG1,$REG3     ; 2ND WORD PATTERN CHECK?
3412  012636  001401                          BEQ     65$             ;
3413  012640  104064                  64$:    ERROR   64              ; PATTERN DOESNT MATCH
3414
3415  012642                          65$:
3416  012642  000402                          BR      TST153          ;;
3417
3418  012644  170360  170360          ADRF7:  .WORD   ALT4N,ALT4N     ; TEST PATTERN
3419
3420
3421                                  ;;**************************************************************
3422                                  ;*TEST 153      TEST OF LOAD-FSRC MODE-4, STORE-FDST MODE-5, F MODE
3423                                  ;;**************************************************************
3424  012650  000004                  TST153: SCOPE
3425  012652  170127  047400                  LDFPS   #047400         ; F MODE FPS
3426  012656  016767  000076  166304          MOV     ADRF10+0,$REG0  ; MOVE PATTERN
3427  012664  016767  000072  166300          MOV     ADRF10+2,$REG1  ;
3428
3429  012672  012705  001174                  MOV     #$REG0+4,R5     ; ADDR(DATA+4)
```

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

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

```
3486  013066  104117  000000          ADRF11: .WORD   ERROR117,0      ; TEST PATTERN
3487
3488
3489                                  ;;*****************************************************************
3490                                  ;*TEST 155      TEST OF LOAD-FSRC MODE-6(PC), STORE-FDST MODE-6(PC), F MODE
3491                                  ;;*****************************************************************
3492  013072  000004                  TST155: SCOPE
3493  013074  170127  047400                  LDFPS   #047400         ; F MODE FPS
3494  013100  016767  000044  166062          MOV     ADRF12+0,$REG0  ; MOVE PATTERN
3495  013106  016767  000040  166056          MOV     ADRF12+2,$REG1  ;
3496
3497  013114  172567  166050                  LDF     $REG0,AC1       ; M6-R7
3498
3499  013120  174167  166050                  STF     AC1,$REG2       ; M6-R7
3500
3501  013124  026767  166040  166042          CMP     $REG0,$REG2     ; 1ST WORD PATTERN CHECK?
3502  013132  001004                          BNE     64$             ;
3503  013134  026767  166032  166034          CMP     $REG1,$REG3     ; 2ND WORD PATTERN CHECK?
3504  013142  001401                          BEQ     65$             ;
3505  013144  104073                  64$:    ERROR   73              ; PATTERN DOESNT MATCH
3506
3507  013146                          65$:
3508  013146  000402                          BR      TST156          ;;
3509
3510  013150  052525  052525          ADRF12: .WORD   ALTP,ALTP       ; TEST PATTERN
3511
3512
3513                                  ;;*****************************************************************
3514                                  ;*TEST 156      TEST OF LOAD-FSRC MODE-0, STORE-FDST MODE-2(PC), F MODE
3515                                  ;;*****************************************************************
3516  013154  000004                  TST156: SCOPE
3517  013156  170127  047400                  LDFPS   #047400         ; F MODE FPS
3518  013162  016767  000062  166000          MOV     ADRF13+0,$REG0  ; MOVE PATTERN
3519  013170  016767  000056  165774          MOV     ADRF13+2,$REG1  ;
3520
3521  013176  172667  165766                  LDF     $REG0,AC2       ; M6-R7 WORKS
3522  013202  172702                          LDF     AC2,AC3         ; M0-R2
3523
3524  013204  012767  104117  000002          MOV     #ERROR117,1$    ; SETUP ERROR CALL FOR WRONG INCREMENT
3525  013212  174327                          STF     AC3,(PC)+       ; M2-R7
3526  013214  000000                  1$:     .WORD   0               ; DEST
3527  013216  000403                          BR      2$              ; THIS SHOULD BE NEXT INSTR AFTER FPP PC MODE 2
3528  013220  104120                          ERROR   120             ; NOT HERE  (+4 INCRE)
3529  013222  104121                          ERROR   121             ; OR HERE  (+6 INCRE)
3530  013224  104122                          ERROR   122             ; OR HERE  (+10 INCRE)
3531  013226  016767  177762  165740  2$:     MOV     1$,$REG2        ; GET DEST
3532
3533  013234  026767  165730  165732          CMP     $REG0,$REG2     ; WAS 1 WORD STORED OK?
3534  013242  001401                          BEQ     64$             ;
3535  013244  104073                          ERROR   73              ; NOT EQUAL, SIGNAL ERROR
3536  013246                          64$:
3537
3538  013246  000402                          BR      TST157          ;;
3539
3540  013250  007417  000000          ADRF13: .WORD   ALT4P,0         ; TEST PATTERN
3541
```

```
3542
3543                                    ;;*****************************************************************
3544                                    ;*TEST 157      TEST OF LOAD-FSRC MODE-6, STORE-FDST MODE-0, F MODE
3545                                    ;;*****************************************************************
3546  013254  000004                   TST157: SCOPE
3547  013256  170127  047400                   LDFPS   #047400          ; F MODE FPS
3548  013262  016767  000052  165700          MOV     ADRF14+0,$REG0   ; MOVE PATTERN
3549  013270  016767  000046  165674          MOV     ADRF14+2,$REG1   ;
3550
3551  013276  012701  001200                   MOV     #$REG0+10,R1     ; ADDR(DATA)+10
3552  013302  172461  177770                   LDF     -10(R1),AC0      ; M6-R1
3553
3554  013306  174001                           STF     AC0,AC1          ; M0-R1
3555  013310  174167  165660                   STF     AC1,$REG2        ; M6-R7 WORKS
3556
3557  013314  026767  165650  165652          CMP     $REG0,$REG2      ; 1ST WORD PATTERN CHECK?
3558  013322  001004                           BNE     64$              ;
3559  013324  026767  165642  165644          CMP     $REG1,$REG3      ; 2ND WORD PATTERN CHECK?
3560  013332  001401                           BEQ     65$              ;
3561  013334  104074                   64$:    ERROR   74               ; PATTERN DOESNT MATCH
3562
3563  013336                           65$:
3564  013336  000402                           BR      TST160           ;;
3565
3566  013340  007417  007417           ADRF14: .WORD   ALT4P,ALT4P      ; TEST PATTERN
3567
3568
```

```
3569
3570                                    ;;*****************************************************************
3571                                            .ENABL  AMA             ; BEGIN ASSEMBLING RELATIVE REFERENCES AS ABSOLUTE
3572                                    ;;*****************************************************************
3573
3574                                    ;;*****************************************************************
3575                                    ;*TEST 160      TEST ALL FP-ACCUM ARE THERE
3576                                    ;;*****************************************************************
3577  013344  000004                   TST160: SCOPE
3578  013346  170127  047400                   LDFPS   #047400         ; INITIAL FPS (SINGLE FLOAT, INTEGER)
3579  013352  005037  001170                   CLR     $REG0           ; SET AC0 = 0,
3580  013356  012737  000005  001172           MOV     #5,$REG1        ;     AC1 = 1,
3581  013364  172437  001170                   LDF     $REG0,AC0       ;     AC2 = 2,
3582  013370  174005                           STF     AC0,AC5         ;     AC3 = 3,
3583  013372  005337  001172                   DEC     $REG1           ;     AC4 = 4,
3584  013376  172437  001170                   LDF     $REG0,AC0       ;     AC5 = 5
3585  013402  174004                           STF     AC0,AC4         ; ...
3586  013404  005337  001172                   DEC     $REG1           ; ...
3587  013410  172737  001170                   LDF     $REG0,AC3       ; ...
3588  013414  005337  001172                   DEC     $REG1           ; ...
3589  013420  172637  001170                   LDF     $REG0,AC2       ; ...
3590  013424  005337  001172                   DEC     $REG1           ; ...
3591  013430  172537  001170                   LDF     $REG0,AC1       ; ...
3592  013434  005337  001172                   DEC     $REG1           ; ...
3593  013440  172437  001170                   LDF     $REG0,AC0       ; ...
3594  013444  005037  001170                   CLR     $REG0           ; NOW TEST THE ABOVE IS TRUE:
3595  013450  005037  001172                   CLR     $REG1           ; ...
3596  013454  174037  001174                   STF     AC0,$REG2       ; DOES AC0 = 0 ?
3597  013460  005737  001174                   TST     $REG2           ; ...
3598  013464  001004                           BNE     1$              ; ...
3599  013466  023737  001172  001176           CMP     $REG1,$REG3     ; ...
3600  013474  001401                           BEQ     11$             ; ...
3601  013476  104075                   1$:     ERROR   75              ;   NO, ERROR
3602  013500  005237  001172           11$:    INC     $REG1           ;   YES, INC FOR NEXT TEST
3603  013504  174137  001174                   STF     AC1,$REG2       ; DOES AC1 = 1 ?
3604  013510  005737  001174                   TST     $REG2           ; ...
3605  013514  001004                           BNE     2$              ; ...
3606  013516  023737  001172  001176           CMP     $REG1,$REG3     ; ...
3607  013524  001401                           BEQ     21$             ; ...
3608  013526  104075                   2$:     ERROR   75              ;   NO, ERROR
3609  013530  005237  001172           21$:    INC     $REG1           ;   YES, INC FOR NEXT TEST
3610  013534  174237  001174                   STF     AC2,$REG2       ; DOES AC2 = 2 ?
3611  013540  005737  001174                   TST     $REG2           ; ...
3612  013544  001004                           BNE     3$              ; ...
3613  013546  023737  001172  001176           CMP     $REG1,$REG3     ; ...
3614  013554  001401                           BEQ     31$             ; ...
3615  013556  104075                   3$:     ERROR   75              ;   NO, ERROR
3616  013560  005237  001172           31$:    INC     $REG1           ;   YES, INC FOR NEXT TEST
3617  013564  174337  001174                   STF     AC3,$REG2       ; DOES AC3 = 3 ?
3618  013570  005737  001174                   TST     $REG2           ; ...
3619  013574  001004                           BNE     4$              ; ...
3620  013576  023737  001172  001176           CMP     $REG1,$REG3     ; ...
3621  013604  001401                           BEQ     41$             ; ...
3622  013606  104075                   4$:     ERROR   75              ;   NO, ERROR
3623  013610  005237  001172           41$:    INC     $REG1           ;   YES, INC FOR NEXT TEST
3624  013614  172404                           LDF     AC4,AC0         ; DOES AC4 = 4 ?
```

```
3625  013616  174037  001174          STF     AC0,$REG2       ; ...
3626  013622  005737  001174          TST     $REG2           ; ...
3627  013626  001004                  BNE     5$              ; ...
3628  013630  023737  001172  001176  CMP     $REG1,$REG3     ; ...
3629  013636  001401                  BEQ     51$             ; ...
3630  013640  104075          5$:     ERROR   75              ;   NO, ERROR
3631  013642  005237  001172  51$:    INC     $REG1           ;   YES, INC FOR NEXT TEST
3632  013646  172405                  LDF     AC5,AC0         ; DOES AC5 = 5 ?
3633  013650  174037  001174          STF     AC0,$REG2       ; ...
3634  013654  005737  001174          TST     $REG2           ; ...
3635  013660  001004                  BNE     6$              ; ...
3636  013662  023737  001172  001176  CMP     $REG1,$REG3     ; ...
3637  013670  001401                  BEQ     61$             ; ...
3638  013672  104075          6$:     ERROR   75              ;   NO, ERROR
3639  013674                  61$:                            ;   YES, GO FOR NEXT TEST
```

```
                                ;;*****************************************************************
                                ;*TEST 161       TEST OF ABSF INSTR, DATA SET ABSF-1
                                ;*              TRUNCATE MODE, ALL INTERRUPT ENABLES ON
                                ;;*****************************************************************
013674  000004                  TST161: SCOPE
013676  012705  013710                  MOV     #ABSF1,R5       ; PTR TO TEST DATA SET
013702  004737  021376                  JSR     PC,@#ABSFT      ; GO TEST

013706  000407                          BR      TST162          ;;

013710                          ABSF1:  ; TEST DATA SET ABSF-1:
013710  000000  000000                  .WORD   0,0     ; INITIAL MEM FLOAT NUMBER
013714  000000  000000                  .WORD   0,0     ; EXPECTED FLOAT RESULT
013720  047453  047444                  .WORD   047453,047444   ; FPS:  BEFORE, AFTER
013724  000000                          .WORD   000000          ; FEC AFTER ( 0 = N/A )


                                ;;*****************************************************************
                                ;*TEST 162       TEST OF ABSF INSTR, DATA SET ABSF-2
                                ;*              TRUNCATE MODE, ALL INTERRUPT ENABLES ON
                                ;;*****************************************************************
013726  000004                  TST162: SCOPE
013730  012705  013742                  MOV     #ABSF2,R5       ; PTR TO TEST DATA SET
013734  004737  021376                  JSR     PC,@#ABSFT      ; GO TEST

013740  000407                          BR      TST163          ;;

013742                          ABSF2:  ; TEST DATA SET ABSF-2:
013742  052525  052525                  .WORD   ALTP,ALTP       ; INITIAL MEM FLOAT NUMBER
013746  052525  052525                  .WORD   ALTP,ALTP       ; EXPECTED FLOAT RESULT
013752  047557  047540                  .WORD   047557,047540   ; FPS:  BEFORE, AFTER
013756  000000                          .WORD   000000          ; FEC AFTER ( 0 = N/A )


                                ;;*****************************************************************
                                ;*TEST 163       TEST OF ABSF INSTR, DATA SET ABSF-3
                                ;*              ROUNDING MODE, ALL INTERRUPT ENABLES ON
                                ;;*****************************************************************
013760  000004                  TST163: SCOPE
013762  012705  013774                  MOV     #ABSF3,R5       ; PTR TO TEST DATA SET
013766  004737  021376                  JSR     PC,@#ABSFT      ; GO TEST

013772  000407                          BR      TST164          ;;

013774                          ABSF3:  ; TEST DATA SET ABSF-3:
013774  125252  125252                  .WORD   NLTN,NLTN       ; INITIAL MEM FLOAT NUMBER
014000  025252  125252                  .WORD   025252,NLTN     ; EXPECTED FLOAT RESULT
014004  047417  047400                  .WORD   047417,047400   ; FPS:  BEFORE, AFTER
014010  000000                          .WORD   000000          ; FEC AFTER ( 0 = N/A )


                                ;;*****************************************************************
                                ;*TEST 164       TEST OF ABSF INSTR, DATA SET ABSF-4
                                ;*              ROUNDING MODE, ALL INTERRUPT ENABLES ON
                                ;;*****************************************************************
014012  000004                  TST164: SCOPE
```

```
3696  014014  012705  014026                MOV     #ABSF4,R5       ; PTR TO TEST DATA SET
3697  014020  004737  021376                JSR     PC,@#ABSFT      ; GO TEST
3698
3699  014024  000407                        BR      TST165          ;;
3700
3701  014026                        ABSF4:  ; TEST DATA SET ABSF-4:
3702  014026  177777  177777                .WORD   LGN,MI  ; INITIAL MEM FLOAT NUMBER
3703  014032  077777  177777                .WORD   LGP,MI  ; EXPECTED FLOAT RESULT
3704  014036  047517  047500                .WORD   047517,047500   ; FPS:  BEFORE, AFTER
3705  014042  000000                        .WORD   000000          ; FEC AFTER ( 0 = N/A )
3706
3707
3708                                ;;**********************************************************************
3709                                ;*TEST 165      TEST OF ABSF INSTR, DATA SET ABSF-5
3710                                ;*             TRUNCATE MODE, ALL INTERRUPT ENABLES ON
3711                                ;;**********************************************************************
3712  014044  000004                TST165: SCOPE
3713  014046  012705  014060                MOV     #ABSF5,R5       ; PTR TO TEST DATA SET
3714  014052  004737  021376                JSR     PC,@#ABSFT      ; GO TEST
3715
3716  014056  000407                        BR      TST166          ;;
3717
3718  014060                        ABSF5:  ; TEST DATA SET ABSF-5:
3719  014060  000200  000000                .WORD   SMP,0   ; INITIAL MEM FLOAT NUMBER
3720  014064  000200  000000                .WORD   SMP,0   ; EXPECTED FLOAT RESULT
3721  014070  047457  047440                .WORD   047457,047440   ; FPS:  BEFORE, AFTER
3722  014074  000000                        .WORD   000000          ; FEC AFTER ( 0 = N/A )
3723
3724
3725                                ;;**********************************************************************
3726                                ;*TEST 166      TEST OF ABSF INSTR, DATA SET ABSF-6
3727                                ;*             TRUNCATE MODE, ALL INTERRUPT ENABLES ON
3728                                ;;**********************************************************************
3729  014076  000004                TST166: SCOPE
3730  014100  012705  014112                MOV     #ABSF6,R5       ; PTR TO TEST DATA SET
3731  014104  004737  021376                JSR     PC,@#ABSFT      ; GO TEST
3732
3733  014110  000407                        BR      TST167          ;;
3734
3735  014112                        ABSF6:  ; TEST DATA SET ABSF-6:
3736  014112  077777  177777                .WORD   LGP,MI  ; INITIAL MEM FLOAT NUMBER
3737  014116  077777  177777                .WORD   LGP,MI  ; EXPECTED FLOAT RESULT
3738  014122  047557  047540                .WORD   047557,047540   ; FPS:  BEFORE, AFTER
3739  014126  000000                        .WORD   000000          ; FEC AFTER ( 0 = N/A )
3740
3741
3742                                ;;**********************************************************************
3743                                ;*TEST 167      TEST OF ABSF INSTR, DATA SET ABSF-7
3744                                ;*             ROUNDING MODE, ALL INTERRUPT ENABLES ON
3745                                ;;**********************************************************************
3746  014130  000004                TST167: SCOPE
3747  014132  012705  014144                MOV     #ABSF7,R5       ; PTR TO TEST DATA SET
3748  014136  004737  021376                JSR     PC,@#ABSFT      ; GO TEST
3749
3750  014142  000407                        BR      TST170          ;;
3751
```

```
3752  014144                          ABSF7:  ; TEST DATA SET ABSF-7:
3753  014144  100200  000000                  .WORD   SMN,0   ; INITIAL MEM FLOAT NUMBER
3754  014150  000200  000000                  .WORD   SMP,0   ; EXPECTED FLOAT RESULT
3755  014154  047417  047400                  .WORD   047417,047400   ; FPS:  BEFORE, AFTER
3756  014160  000000                          .WORD   000000          ; FEC AFTER ( 0 = N/A )
3757
3758
3759                                  ;;*****************************************************************
3760                                  ;*TEST 170       TEST OF ABSF INSTR, DATA SET ABSF-10
3761                                  ;*              ROUNDING MODE, ALL INTERRUPT ENABLES ON
3762                                  ;;*****************************************************************
3763  014162  000004                  TST170: SCOPE
3764  014164  012705  014176                  MOV     #ABSF10,R5      ; PTR TO TEST DATA SET
3765  014170  004737  021376                  JSR     PC,@#ABSFT      ; GO TEST
3766
3767  014174  000407                          BR      TST171          ;;
3768
3769  014176                          ABSF10: ; TEST DATA SET ABSF-10:
3770  014176  000177  177777                  .WORD   ZX1MP,M1        ; INITIAL MEM FLOAT NUMBER
3771  014202  000000  000000                  .WORD   0,0     ; EXPECTED FLOAT RESULT
3772  014206  047513  047504                  .WORD   047513,047504   ; FPS:  BEFORE, AFTER
3773  014212  000000                          .WORD   000000          ; FEC AFTER ( 0 = N/A )
3774
3775
3776                                  ;;*****************************************************************
3777                                  ;*TEST 171       TEST OF ABSF INSTR, DATA SET ABSF-11
3778                                  ;*              TRUNCATE MODE, ALL INTERRUPT ENABLES ON
3779                                  ;;*****************************************************************
3780  014214  000004                  TST171: SCOPE
3781  014216  012705  014230                  MOV     #ABSF11,R5      ; PTR TO TEST DATA SET
3782  014222  004737  021376                  JSR     PC,@#ABSFT      ; GO TEST
3783
3784  014226  000407                          BR      TST172          ;;
3785
3786  014230                          ABSF11: ; TEST DATA SET ABSF-11:
3787  014230  100000  000000                  .WORD   M0,0    ; INITIAL MEM FLOAT NUMBER
3788  014234  000000  000000                  .WORD   0,0     ; EXPECTED FLOAT RESULT
3789  014240  047453  147444                  .WORD   047453,147444   ; FPS:  BEFORE, AFTER
3790  014244  100014                          .WORD   100014          ; FEC AFTER ( 0 = N/A )
3791
3792
3793                                  ;;*****************************************************************
3794                                  ;*TEST 172       TEST OF ABSF INSTR, DATA SET ABSF-12
3795                                  ;*              TRUNCATE MODE, ALL INTERRUPT ENABLES ON
3796                                  ;;*****************************************************************
3797  014246  000004                  TST172: SCOPE
3798  014250  012705  014262                  MOV     #ABSF12,R5      ; PTR TO TEST DATA SET
3799  014254  004737  021376                  JSR     PC,@#ABSFT      ; GO TEST
3800
3801  014260  000407                          BR      TST173          ;;
3802
3803  014262                          ABSF12: ; TEST DATA SET ABSF-12:
3804  014262  100000  000001                  .WORD   M0,1    ; INITIAL MEM FLOAT NUMBER
3805  014266  000000  000000                  .WORD   0,0     ; EXPECTED FLOAT RESULT
3806  014272  047553  147544                  .WORD   047553,147544   ; FPS:  BEFORE, AFTER
3807  014276  100014                          .WORD   100014          ; FEC AFTER ( 0 = N/A )
```

```
3808
3809
3810                                    ;;*****************************************************************
3811                                    ;*TEST 173       TEST OF ABSF INSTR, DATA SET ABSF-13
3812                                    ;*              ROUNDING MODE, -D INTERRUPT ENABLE OFF, ALL OTHERS ON
3813                                    ;;*****************************************************************
3814  014300  000004                    TST173: SCOPE
3815  014302  012705  014314                    MOV     #ABSF13,R5      ; PTR TO TEST DATA SET
3816  014306  004737  021376                    JSR     PC,@#ABSFT      ; GO TEST
3817
3818  014312  000407                            BR      TST174          ;;
3819
3820  014314                            ABSF13: ; TEST DATA SET ABSF-13:
3821  014314  100177  177777                    .WORD   2X1MN,M1        ; INITIAL MEM FLOAT NUMBER
3822  014320  000000  000000                    .WORD   0,0     ; EXPECTED FLOAT RESULT
3823  014324  043413  043404                    .WORD   043413,043404   ; FPS:  BEFORE, AFTER
3824  014330  000000                            .WORD   000000          ; FEC AFTER ( 0 = N/A )
3825
3826
3827
```

```
                                        ;;*****************************************************************
                                        ;*TEST 174       TEST OF ABSD INSTR, DATA SET ABSD-1
                                        ;*              ROUNDING MODE, ALL INTERRUPT ENABLES ON
                                        ;;*****************************************************************
014332  000004                        TST174: SCOPE
014334  012705  014346                        MOV     #ABSD1,R5       ; PTR TO TEST DATA SET
014340  004737  021552                        JSR     PC,@#ABSDT      ; GO TEST
014344  000413                              BR      TST175          ;;
014346                                  ABSD1:  ; TEST DATA SET ABSD-1:
014346  000000  000000  000000                .WORD   0,0,0,0 ; INITIAL MEM FLOAT NUMBER
014354  000000
014356  000000  000000  000000                .WORD   0,0,0,0 ; EXPECTED FLOAT RESULT
014364  000000
014366  047713  047704                        .WORD   047713,047704   ; FPS:  BEFORE, AFTER
014372  000000                                .WORD   000000          ; FEC AFTER ( 0 = N/A )
                                        ;;*****************************************************************
                                        ;*TEST 175       TEST OF ABSD INSTR, DATA SET ABSD-2
                                        ;*              TRUNCATE MODE, ALL INTERRUPT ENABLES ON
                                        ;;*****************************************************************
014374  000004                        TST175: SCOPE
014376  012705  014410                        MOV     #ABSD2,R5       ; PTR TO TEST DATA SET
014402  004737  021552                        JSR     PC,@#ABSDT      ; GO TEST
014406  000413                              BR      TST176          ;;
014410                                  ABSD2:  ; TEST DATA SET ABSD-2:
014410  052525  052525  052525                .WORD   ALTP,ALTP,ALTP,ALTP     ; INITIAL MEM FLOAT NUMBER
014416  052525
014420  052525  052525  052525                .WORD   ALTP,ALTP,ALTP,ALTP     ; EXPECTED FLOAT RESULT
014426  052525
014430  047657  047640                        .WORD   047657,047640   ; FPS:  BEFORE, AFTER
014434  000000                                .WORD   000000          ; FEC AFTER ( 0 = N/A )
                                        ;;*****************************************************************
                                        ;*TEST 176       TEST OF ABSD INSTR, DATA SET ABSD-3
                                        ;*              TRUNCATE MODE, ALL INTERRUPT ENABLES ON
                                        ;;*****************************************************************
014436  000004                        TST176: SCOPE
014440  012705  014452                        MOV     #ABSD3,R5       ; PTR TO TEST DATA SET
014444  004737  021552                        JSR     PC,@#ABSDT      ; GO TEST
014450  000413                              BR      TST177          ;;
014452                                  ABSD3:  ; TEST DATA SET ABSD-3:
014452  125252  125252  125252                .WORD   ALTN,ALTN,ALTN,ALTN     ; INITIAL MEM FLOAT NUMBER
014460  125252
014462  025252  125252  125252                .WORD   025252,ALTN,ALTN,ALTN   ; EXPECTED FLOAT RESULT
014470  125252
014472  047757  047740                        .WORD   047757,047740   ; FPS:  BEFORE, AFTER
014476  000000                                .WORD   000000          ; FEC AFTER ( 0 = N/A )
```

```
                                        ;;*****************************************************************
                                        ;*TEST 177       TEST OF ABSD INSTR, DATA SET ABSD-4
                                        ;*              ROUNDING MODE, ALL INTERRUPT ENABLES ON
                                        ;;*****************************************************************
014500  000004                          TST177: SCOPE
014502  012705  014514                          MOV     #ABSD4,R5       ; PTR TO TEST DATA SET
014506  004737  021552                          JSR     PC,@#ABSDT      ; GO TEST

014512  000413                          BR      TST200          ;;

014514                                  ABSD4:  ; TEST DATA SET ABSD-4:
014514  177777  177777  177777                  .WORD   LGN,M1,M1,M1    ; INITIAL MEM FLOAT NUMBER
014522  177777
014524  077777  177777  177777                  .WORD   LGP,M1,M1,M1    ; EXPECTED FLOAT RESULT
014532  177777
014534  047617  047600                          .WORD   047617,047600   ; FPS:  BEFORE, AFTER
014540  000000                                  .WORD   000000          ; FEC AFTER ( 0 = N/A )


                                        ;;*****************************************************************
                                        ;*TEST 200       TEST OF ABSD INSTR, DATA SET ABSD-5
                                        ;*              ROUNDING MODE, ALL INTERRUPT ENABLES ON
                                        ;;*****************************************************************
014542  000004                          TST200: SCOPE
014544  012705  014556                          MOV     #ABSD5,R5       ; PTR TO TEST DATA SET
014550  004737  021552                          JSR     PC,@#ABSDT      ; GO TEST

014554  000413                          BR      TST201          ;;

014556                                  ABSD5:  ; TEST DATA SET ABSD-5:
014556  000200  000000  000000                  .WORD   SMP,0,0,0       ; INITIAL MEM FLOAT NUMBER
014564  000000
014566  000200  000000  000000                  .WORD   SMP,0,0,0       ; EXPECTED FLOAT RESULT
014574  000000
014576  047717  047700                          .WORD   047717,047700   ; FPS:  BEFORE, AFTER
014602  000000                                  .WORD   000000          ; FEC AFTER ( 0 = N/A )


                                        ;;*****************************************************************
                                        ;*TEST 201       TEST OF ABSD INSTR, DATA SET ABSD-6
                                        ;*              TRUNCATE MODE, ALL INTERRUPT ENABLES ON
                                        ;;*****************************************************************
014604  000004                          TST201: SCOPE
014606  012705  014620                          MOV     #ABSD6,R5       ; PTR TO TEST DATA SET
014612  004737  021552                          JSR     PC,@#ABSDT      ; GO TEST

014616  000413                          BR      TST202          ;;

014620                                  ABSD6:  ; TEST DATA SET ABSD-6:
014620  077777  177777  177777                  .WORD   LGP,M1,M1,M1    ; INITIAL MEM FLOAT NUMBER
014626  177777
014630  077777  177777  177777                  .WORD   LGP,M1,M1,M1    ; EXPECTED FLOAT RESULT
014636  177777
014640  047657  047640                          .WORD   047657,047640   ; FPS:  BEFORE, AFTER
014644  000000                                  .WORD   000000          ; FEC AFTER ( 0 = N/A )
```

```
                                ;;*****************************************************************
                                ;*TEST 202       TEST OF ABSD INSTR, DATA SET ABSD-7
                                ;*              TRUNCATE MODE, ALL INTERRUPT ENABLES ON
                                ;;*****************************************************************
014646  000004                  TST202: SCOPE
014650  012705  014662                  MOV     #ABSD7,R5       ; PTR TO TEST DATA SET
014654  004737  021552                  JSR     PC,@#ABSDT      ; GO TEST
                                
014660  000413                          BR      TST203          ;;
                                
014662                          ABSD7:  ; TEST DATA SET ABSD-7:
014662  100200  000000  000000          .WORD   SMN,0,0,0       ; INITIAL MEM FLOAT NUMBER
014670  000000
014672  000200  000000  000000          .WORD   SMP,0,0,0       ; EXPECTED FLOAT RESULT
014700  000000
014702  047757  047740                  .WORD   047757,047740   ; FPS:  BEFORE, AFTER
014706  000000                          .WORD   000000          ; FEC AFTER ( 0 = N/A )
                                
                                
                                ;;*****************************************************************
                                ;*TEST 203       TEST OF ABSD INSTR, DATA SET ABSD-10
                                ;*              ROUNDING MODE, ALL INTERRUPT ENABLES ON
                                ;;*****************************************************************
014710  000004                  TST203: SCOPE
014712  012705  014724                  MOV     #ABSD10,R5      ; PTR TO TEST DATA SET
014716  004737  021552                  JSR     PC,@#ABSDT      ; GO TEST
                                
014722  000413                          BR      TST204          ;;
                                
014724                          ABSD10: ; TEST DATA SET ABSD-10:
014724  000177  177777  177777          .WORD   ZX1MP,M1,M1,M1  ; INITIAL MEM FLOAT NUMBER
014732  177777
014734  000000  000000  000000          .WORD   0,0,0,0 ; EXPECTED FLOAT RESULT
014742  000000
014744  047613  047604                  .WORD   047613,047604   ; FPS:  BEFORE, AFTER
014750  000000                          .WORD   000000          ; FEC AFTER ( 0 = N/A )
                                
                                
                                ;;*****************************************************************
                                ;*TEST 204       TEST OF ABSD INSTR, DATA SET ABSD-11
                                ;*              ROUNDING MODE, ALL INTERRUPT ENABLES ON
                                ;;*****************************************************************
014752  000004                  TST204: SCOPE
014754  012705  014766                  MOV     #ABSD11,R5      ; PTR TO TEST DATA SET
014760  004737  021552                  JSR     PC,@#ABSDT      ; GO TEST
                                
014764  000413                          BR      TST205          ;;
                                
014766                          ABSD11: ; TEST DATA SET ABSD-11:
014766  100000  000000  000000          .WORD   M0,0,0,0        ; INITIAL MEM FLOAT NUMBER
014774  000000
014776  000000  000000  000000          .WORD   0,0,0,0 ; EXPECTED FLOAT RESULT
015004  000000
015006  047713  147704                  .WORD   047713,147704   ; FPS:  BEFORE, AFTER
```

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

 3996  015012  100014                          .WORD   100014          ; FEC AFTER ( 0 = N/A )
 3997
 3998
 3999                                  ;;*****************************************************************
 4000                                  ;*TEST 205       TEST OF ABSD INSTR, DATA SET ABSD-12
 4001                                  ;*              TRUNCATE MODE, ALL INTERRUPT ENABLES ON
 4002                                  ;;*****************************************************************
 4003  015014  000004                  TST205: SCOPE
 4004  015016  012705  015030                  MOV     #ABSD12,R5      ; PTR TO TEST DATA SET
 4005  015022  004737  021552                  JSR     PC,@#ABSDT      ; GO TEST
 4006
 4007  015026  000413                          BR      TST206          ;;
 4008
 4009  015030                          ABSD12: ; TEST DATA SET ABSD-12:
 4010  015030  100000  000000  000000          .WORD   M0,0,0,1        ; INITIAL MEM FLOAT NUMBER
 4011  015036  000001
 4012  015040  000000  000000  000000          .WORD   0,0,0,0 ; EXPECTED FLOAT RESULT
 4013  015046  000000
 4014  015050  047653  147644                  .WORD   047653,147644   ; FPS:  BEFORE, AFTER
 4015  015054  100014                          .WORD   100014          ; FEC AFTER ( 0 = N/A )
 4016
 4017
 4018                                  ;;*****************************************************************
 4019                                  ;*TEST 206       TEST OF ABSD INSTR, DATA SET ABSD-13
 4020                                  ;*              TRUNCATE MODE, -0 INTERRUPT ENABLE OFF, ALL OTHERS ON
 4021                                  ;;*****************************************************************
 4022  015056  000004                  TST206: SCOPE
 4023  015060  012705  015072                  MOV     #ABSD13,R5      ; PTR TO TEST DATA SET
 4024  015064  004737  021552                  JSR     PC,@#ABSDT      ; GO TEST
 4025
 4026  015070  000413                          BR      TST207          ;;
 4027
 4028  015072                          ABSD13: ; TEST DATA SET ABSD-13:
 4029  015072  100177  177777  177777          .WORD   ZX1MN,M1,M1,M1  ; INITIAL MEM FLOAT NUMBER
 4030  015100  177777
 4031  015102  000000  000000  000000          .WORD   0,0,0,0 ; EXPECTED FLOAT RESULT
 4032  015110  000000
 4033  015112  043653  043644                  .WORD   043653,043644   ; FPS:  BEFORE, AFTER
 4034  015116  000000                          .WORD   000000          ; FEC AFTER ( 0 = N/A )
 4035
 4036
 4037
```

```
4038                                    ;;*****************************************************************
4039                                    ;*TEST 207      TEST OF NEGF INSTR, DATA SET NEGF-1
4040                                    ;*             ROUNDING MODE, ALL INTERRUPT ENABLES ON
4041                                    ;;*****************************************************************
4042  015120  000004                    TST207: SCOPE
4043  015122  012705  015134                    MOV     #NEGF1,R5       ; PTR TO TEST DATA SET
4044  015126  004737  021762                    JSR     PC,@#NEGFT      ; GO TEST
4045
4046  015132  000407                            BR      TST210          ;;
4047
4048  015134                          NEGF1:  ; TEST DATA SET NEGF-1:
4049  015134  000000  000000                    .WORD   0,0     ; INITIAL MEM FLOAT NUMBER
4050  015140  000000  000000                    .WORD   0,0     ; EXPECTED FLOAT RESULT
4051  015144  047413  047404                    .WORD   047413,047404   ; FPS:  BEFORE, AFTER
4052  015150  000000                            .WORD   000000          ; FEC AFTER ( 0 = N/A )
4053
4054
4055                                    ;;*****************************************************************
4056                                    ;*TEST 210      TEST OF NEGF INSTR, DATA SET NEGF-2
4057                                    ;*             TRUNCATE MODE, ALL INTERRUPT ENABLES ON
4058                                    ;;*****************************************************************
4059  015152  000004                    TST210: SCOPE
4060  015154  012705  015166                    MOV     #NEGF2,R5       ; PTR TO TEST DATA SET
4061  015160  004737  021762                    JSR     PC,@#NEGFT      ; GO TEST
4062
4063  015164  000407                            BR      TST211          ;;
4064
4065  015166                          NEGF2:  ; TEST DATA SET NEGF-2:
4066  015166  052525  052525                    .WORD   ALTP,ALTP       ; INITIAL MEM FLOAT NUMBER
4067  015172  152525  052525                    .WORD   152525,ALTP     ; EXPECTED FLOAT RESULT
4068  015176  047547  047550                    .WORD   047547,047550   ; FPS:  BEFORE, AFTER
4069  015202  000000                            .WORD   000000          ; FEC AFTER ( 0 = N/A )
4070
4071
4072                                    ;;*****************************************************************
4073                                    ;*TEST 211      TEST OF NEGF INSTR, DATA SET NEGF-3
4074                                    ;*             ROUNDING MODE, ALL INTERRUPT ENABLES ON
4075                                    ;;*****************************************************************
4076  015204  000004                    TST211: SCOPE
4077  015206  012705  015220                    MOV     #NEGF3,R5       ; PTR TO TEST DATA SET
4078  015212  004737  021762                    JSR     PC,@#NEGFT      ; GO TEST
4079
4080  015216  000407                            BR      TST212          ;;
4081
4082  015220                          NEGF3:  ; TEST DATA SET NEGF-3:
4083  015220  125252  125252                    .WORD   ALTN,ALTN       ; INITIAL MEM FLOAT NUMBER
4084  015224  025252  125252                    .WORD   025252,ALTN     ; EXPECTED FLOAT RESULT
4085  015230  047517  047500                    .WORD   047517,047500   ; FPS:  BEFORE, AFTER
4086  015234  000000                            .WORD   000000          ; FEC AFTER ( 0 = N/A )
4087
4088
4089                                    ;;*****************************************************************
4090                                    ;*TEST 212      TEST OF NEGF INSTR, DATA SET NEGF-4
4091                                    ;*             TRUNCATE MODE, ALL INTERRUPT ENABLES ON
4092                                    ;;*****************************************************************
4093  015236  000004                    TST212: SCOPE
```

```
4094  015240  012705  015252          MOV     #NEGF4,R5       ; PTR TO TEST DATA SET
4095  015244  004737  021762          JSR     PC,@#NEGFT      ; GO TEST
4096
4097  015250  000407                  BR      TST213          ;;
4098
4099  015252                  NEGF4:  ; TEST DATA SET NEGF-4:
4100  015252  177777  177777          .WORD   LGN,M1  ; INITIAL MEM FLOAT NUMBER
4101  015256  077777  177777          .WORD   LGP,M1  ; EXPECTED FLOAT RESULT
4102  015262  047457  047440          .WORD   047457,047440   ; FPS:  BEFORE, AFTER
4103  015266  000000                  .WORD   000000          ; FEC AFTER ( 0 = N/A )
4104
4105
4106                          ;;**********************************************************************
4107                          ;*TEST 213      TEST OF NEGF INSTR, DATA SET NEGF-5
4108                          ;*             ROUNDING MODE, ALL INTERRUPT ENABLES ON
4109                          ;;**********************************************************************
4110  015270  000004          TST213: SCOPE
4111  015272  012705  015304          MOV     #NEGF5,R5       ; PTR TO TEST DATA SET
4112  015276  004737  021762          JSR     PC,@#NEGFT      ; GO TEST
4113
4114  015302  000407                  BR      TST214          ;;
4115
4116  015304                  NEGF5:  ; TEST DATA SET NEGF-5:
4117  015304  000200  000000          .WORD   SMP,0   ; INITIAL MEM FLOAT NUMBER
4118  015310  100200  000000          .WORD   SMN,0   ; EXPECTED FLOAT RESULT
4119  015314  047407  047410          .WORD   047407,047410   ; FPS:  BEFORE, AFTER
4120  015320  000000                  .WORD   000000          ; FEC AFTER ( 0 = N/A )
4121
4122
4123                          ;;**********************************************************************
4124                          ;*TEST 214      TEST OF NEGF INSTR, DATA SET NEGF-6
4125                          ;*             TRUNCATE MODE, ALL INTERRUPT ENABLES ON
4126                          ;;**********************************************************************
4127  015322  000004          TST214: SCOPE
4128  015324  012705  015336          MOV     #NEGF6,R5       ; PTR TO TEST DATA SET
4129  015330  004737  021762          JSR     PC,@#NEGFT      ; GO TEST
4130
4131  015334  000407                  BR      TST215          ;;
4132
4133  015336                  NEGF6:  ; TEST DATA SET NEGF-6:
4134  015336  077777  177777          .WORD   LGP,M1  ; INITIAL MEM FLOAT NUMBER
4135  015342  177777  177777          .WORD   LGN,M1  ; EXPECTED FLOAT RESULT
4136  015346  047547  047550          .WORD   047547,047550   ; FPS:  BEFORE, AFTER
4137  015352  000000                  .WORD   000000          ; FEC AFTER ( 0 = N/A )
4138
4139
4140                          ;;**********************************************************************
4141                          ;*TEST 215      TEST OF NEGF INSTR, DATA SET NEGF-7
4142                          ;*             ROUNDING MODE, ALL INTERRUPT ENABLES ON
4143                          ;;**********************************************************************
4144  015354  000004          TST215: SCOPE
4145  015356  012705  015370          MOV     #NEGF7,R5       ; PTR TO TEST DATA SET
4146  015362  004737  021762          JSR     PC,@#NEGFT      ; GO TEST
4147
4148  015366  000407                  BR      TST216          ;;
4149
```

```
4150  015370                          NEGF7:  ; TEST DATA SET NEGF-7:
4151  015370  100200  000000                  .WORD   SMN,0   ; INITIAL MEM FLOAT NUMBER
4152  015374  000200  000000                  .WORD   SMP,0   ; EXPECTED FLOAT RESULT
4153  015400  047517  047500                  .WORD   047517,047500   ; FPS:  BEFORE, AFTER
4154  015404  000000                          .WORD   000000          ; FEC AFTER ( 0 = N/A )
4155
4156
4157                                  ;;**********************************************************************
4158                                  ;*TEST 216      TEST OF NEGF INSTR, DATA SET NEGF-10
4159                                  ;*             TRUNCATE MODE, ALL INTERRUPT ENABLES ON
4160                                  ;;**********************************************************************
4161  015406  000004                  TST216: SCOPE
4162  015410  012705  015422                  MOV     #NEGF10,R5      ; PTR TO TEST DATA SET
4163  015414  004737  021762                  JSR     PC,@#NEGFT      ; GO TEST
4164
4165  015420  000407                          BR      TST217          ;;
4166
4167  015422                          NEGF10: ; TEST DATA SET NEGF-10:
4168  015422  000177  177777                  .WORD   ZX1MP,M1        ; INITIAL MEM FLOAT NUMBER
4169  015426  000000  000000                  .WORD   0,0     ; EXPECTED FLOAT RESULT
4170  015432  047453  047444                  .WORD   047453,047444   ; FPS:  BEFORE, AFTER
4171  015436  000000                          .WORD   000000          ; FEC AFTER ( 0 = N/A )
4172
4173
4174                                  ;;**********************************************************************
4175                                  ;*TEST 217      TEST OF NEGF INSTR, DATA SET NEGF-11
4176                                  ;*             TRUNCATE MODE, ALL INTERRUPT ENABLES ON
4177                                  ;;**********************************************************************
4178  015440  000004                  TST217: SCOPE
4179  015442  012705  015454                  MOV     #NEGF11,R5      ; PTR TO TEST DATA SET
4180  015446  004737  021762                  JSR     PC,@#NEGFT      ; GO TEST
4181
4182  015452  000407                          BR      TST220          ;;
4183
4184  015454                          NEGF11: ; TEST DATA SET NEGF-11:
4185  015454  100000  000000                  .WORD   M0,0    ; INITIAL MEM FLOAT NUMBER
4186  015460  000000  000000                  .WORD   0,0     ; EXPECTED FLOAT RESULT
4187  015464  047453  147444                  .WORD   047453,147444   ; FPS:  BEFORE, AFTER
4188  015470  100014                          .WORD   100014          ; FEC AFTER ( 0 = N/A )
4189
4190
4191                                  ;;**********************************************************************
4192                                  ;*TEST 220      TEST OF NEGF INSTR, DATA SET NEGF-12
4193                                  ;*             TRUNCATE MODE, ALL INTERRUPT ENABLES ON
4194                                  ;;**********************************************************************
4195  015472  000004                  TST220: SCOPE
4196  015474  012705  015506                  MOV     #NEGF12,R5      ; PTR TO TEST DATA SET
4197  015500  004737  021762                  JSR     PC,@#NEGFT      ; GO TEST
4198
4199  015504  000407                          BR      TST221          ;;
4200
4201  015506                          NEGF12: ; TEST DATA SET NEGF-12:
4202  015506  100000  000001                  .WORD   M0,1    ; INITIAL MEM FLOAT NUMBER
4203  015512  000000  000000                  .WORD   0,0     ; EXPECTED FLOAT RESULT
4204  015516  047553  147544                  .WORD   047553,147544   ; FPS:  BEFORE, AFTER
4205  015522  100014                          .WORD   100014          ; FEC AFTER ( 0 = N/A )
```

```
                                ;;*****************************************************************
                                ;*TEST 221       TEST OF NEGF INSTR, DATA SET NEGF-13
                                ;*              ROUNDING MODE, -0 INTERRUPT ENABLE OFF, ALL OTHERS ON
                                ;;*****************************************************************
015524  000004                  TST221: SCOPE
015526  012705  015540                  MOV     #NEGF13,R5      ; PTR TO TEST DATA SET
015532  004737  021762                  JSR     PC,@#NEGFT      ; GO TEST
015536  000407                          BR      TST222          ;;
015540                          NEGF13: ; TEST DATA SET NEGF-13:
015540  100177  177777                  .WORD   ZX1MN,M1        ; INITIAL MEM FLOAT NUMBER
015544  000000  000000                  .WORD   0,0             ; EXPECTED FLOAT RESULT
015550  043513  043504                  .WORD   043513,043504   ; FPS:  BEFORE, AFTER
015554  000000                          .WORD   000000          ; FEC AFTER ( 0 = N/A )
```

```
                                ;;*****************************************************************
                                ;*TEST 222       TEST OF NEGD INSTR, DATA SET NEGD-1
                                ;*              TRUNCATE MODE, ALL INTERRUPT ENABLES ON
                                ;;*****************************************************************
015556  000004                TST222: SCOPE
015560  012705  015572                MOV     #NEGD1,R5       ; PTR TO TEST DATA SET
015564  004737  022136                JSR     PC,@#NEGDT      ; GO TEST
015570  000413                        BR      TST223          ;;
015572                          NEGD1:  ; TEST DATA SET NEGD-1:
015572  000000  000000  000000        .WORD   0,0,0,0 ; INITIAL MEM FLOAT NUMBER
015600  000000
015602  000000  000000  000000        .WORD   0,0,0,0 ; EXPECTED FLOAT RESULT
015610  000000
015612  047753  047744                .WORD   047753,047744   ; FPS:  BEFORE, AFTER
015616  000000                        .WORD   000000          ; FEC AFTER ( 0 = N/A )


                                ;;*****************************************************************
                                ;*TEST 223       TEST OF NEGD INSTR, DATA SET NEGD-2
                                ;*              ROUNDING MODE, ALL INTERRUPT ENABLES ON
                                ;;*****************************************************************
015620  000004                TST223: SCOPE
015622  012705  015634                MOV     #NEGD2,R5       ; PTR TO TEST DATA SET
015626  004737  022136                JSR     PC,@#NEGDT      ; GO TEST
015632  000413                        BR      TST224          ;;
015634                          NEGD2:  ; TEST DATA SET NEGD-2:
015634  052525  052525  052525        .WORD   ALTP,ALTP,ALTP,ALTP     ; INITIAL MEM FLOAT NUMBER
015642  052525
015644  152525  052525  052525        .WORD   152525,ALTP,ALTP,ALTP   ; EXPECTED FLOAT RESULT
015652  052525
015654  047607  047610                .WORD   047607,047610   ; FPS:  BEFORE, AFTER
015660  000000                        .WORD   000000          ; FEC AFTER ( 0 = N/A )


                                ;;*****************************************************************
                                ;*TEST 224       TEST OF NEGD INSTR, DATA SET NEGD-3
                                ;*              TRUNCATE MODE, ALL INTERRUPT ENABLES ON
                                ;;*****************************************************************
015662  000004                TST224: SCOPE
015664  012705  015676                MOV     #NEGD3,R5       ; PTR TO TEST DATA SET
015670  004737  022136                JSR     PC,@#NEGDT      ; GO TEST
015674  000413                        BR      TST225          ;;
015676                          NEGD3:  ; TEST DATA SET NEGD-3:
015676  125252  125252  125252        .WORD   ALTN,ALTN,ALTN,ALTN     ; INITIAL MEM FLOAT NUMBER
015704  125252
015706  025252  125252  125252        .WORD   025252,ALTN,ALTN,ALTN   ; EXPECTED FLOAT RESULT
015714  125252
015716  047657  047640                .WORD   047657,047640   ; FPS:  BEFORE, AFTER
015722  000000                        .WORD   000000          ; FEC AFTER ( 0 = N/A )
```

```
                                ;;*****************************************************************
                                ;*TEST 225       TEST OF NEGD INSTR, DATA SET NEGD-4
                                ;*              ROUNDING MODE, ALL INTERRUPT ENABLES ON
                                ;;*****************************************************************
015724  000004                TST225: SCOPE
015726  012705  015740              MOV     #NEGD4,R5       ; PTR TO TEST DATA SET
015732  004737  022136              JSR     PC,@#NEGDT      ; GO TEST
015736  000413                      BR      TST226          ;;
015740                          NEGD4:  ; TEST DATA SET NEGD-4:
015740  177777  177777  177777        .WORD   LGN,M1,M1,M1    ; INITIAL MEM FLOAT NUMBER
015746  177777
015750  077777  177777  177777        .WORD   LGP,M1,M1,M1    ; EXPECTED FLOAT RESULT
015756  177777
015760  047717  047700              .WORD   047717,047700   ; FPS:  BEFORE, AFTER
015764  000000                      .WORD   000000          ; FEC AFTER ( 0 = N/A )

                                ;;*****************************************************************
                                ;*TEST 226       TEST OF NEGD INSTR, DATA SET NEGD-5
                                ;*              TRUNCATE MODE, ALL INTERRUPT ENABLES ON
                                ;;*****************************************************************
015766  000004                TST226: SCOPE
015770  012705  016002              MOV     #NEGD5,R5       ; PTR TO TEST DATA SET
015774  004737  022136              JSR     PC,@#NEGDT      ; GO TEST
016000  000413                      BR      TST227          ;;
016002                          NEGD5:  ; TEST DATA SET NEGD-5:
016002  000200  000000  000000        .WORD   SMP,0,0,0       ; INITIAL MEM FLOAT NUMBER
016010  000000
016012  100200  000000  000000        .WORD   SMN,0,0,0       ; EXPECTED FLOAT RESULT
016020  000000
016022  047747  047750              .WORD   047747,047750   ; FPS:  BEFORE, AFTER
016026  000000                      .WORD   000000          ; FEC AFTER ( 0 = N/A )

                                ;;*****************************************************************
                                ;*TEST 227       TEST OF NEGD INSTR, DATA SET NEGD-6
                                ;*              ROUNDING MODE, ALL INTERRUPT ENABLES ON
                                ;;*****************************************************************
016030  000004                TST227: SCOPE
016032  012705  016044              MOV     #NEGD6,R5       ; PTR TO TEST DATA SET
016036  004737  022136              JSR     PC,@#NEGDT      ; GO TEST
016042  000413                      BR      TST230          ;;
016044                          NEGD6:  ; TEST DATA SET NEGD-6:
016044  077777  177777  177777        .WORD   LGP,M1,M1,M1    ; INITIAL MEM FLOAT NUMBER
016052  177777
016054  177777  177777  177777        .WORD   LGN,M1,M1,M1    ; EXPECTED FLOAT RESULT
016062  177777
016064  047607  047610              .WORD   047607,047610   ; FPS:  BEFORE, AFTER
016070  000000                      .WORD   000000          ; FEC AFTER ( 0 = N/A )
```

```
                                        ;;*****************************************************************
                                        ;*TEST 230      TEST OF NEGD INSTR, DATA SET NEGD-7
                                        ;*             TRUNCATE MODE, ALL INTERRUPT ENABLES ON
                                        ;;*****************************************************************
016072  000004                          TST230: SCOPE
016074  012705  016106                          MOV     #NEGD7,R5       ; PTR TO TEST DATA SET
016100  004737  022136                          JSR     PC,@#NEGDT      ; GO TEST

016104  000413                          BR      TST231          ;;

016106                                  NEGD7:  ; TEST DATA SET NEGD-7:
016106  100200  000000  000000                  .WORD   SMN,0,0,0       ; INITIAL MEM FLOAT NUMBER
016114  000000
016116  000200  000000  000000                  .WORD   SMP,0,0,0       ; EXPECTED FLOAT RESULT
016124  000000
016126  047657  047640                          .WORD   047657,047640   ; FPS:  BEFORE, AFTER
016132  000000                                  .WORD   000000          ; FEC AFTER ( 0 = N/A )


                                        ;;*****************************************************************
                                        ;*TEST 231      TEST OF NEGD INSTR, DATA SET NEGD-10
                                        ;*             ROUNDING MODE, ALL INTERRUPT ENABLES ON
                                        ;;*****************************************************************
016134  000004                          TST231: SCOPE
016136  012705  016150                          MOV     #NEGD10,R5      ; PTR TO TEST DATA SET
016142  004737  022136                          JSR     PC,@#NEGDT      ; GO TEST

016146  000413                          BR      TST232          ;;

016150                                  NEGD10: ; TEST DATA SET NEGD-10:
016150  000177  177777  177777                  .WORD   ZX1MP,M1,M1,M1  ; INITIAL MEM FLOAT NUMBER
016156  177777
016160  000000  000000  000000                  .WORD   0,0,0,0 ; EXPECTED FLOAT RESULT
016166  000000
016170  047713  047704                          .WORD   047713,047704   ; FPS:  BEFORE, AFTER
016174  000000                                  .WORD   000000          ; FEC AFTER ( 0 = N/A )


                                        ;;*****************************************************************
                                        ;*TEST 232      TEST OF NEGD INSTR, DATA SET NEGD-11
                                        ;*             ROUNDING MODE, ALL INTERRUPT ENABLES ON
                                        ;;*****************************************************************
016176  000004                          TST232: SCOPE
016200  012705  016212                          MOV     #NEGD11,R5      ; PTR TO TEST DATA SET
016204  004737  022136                          JSR     PC,@#NEGDT      ; GO TEST

016210  000413                          BR      TST233          ;;

016212                                  NEGD11: ; TEST DATA SET NEGD-11:
016212  100000  000000  000000                  .WORD   M0,0,0,0        ; INITIAL MEM FLOAT NUMBER
016220  000000
016222  000000  000000  000000                  .WORD   0,0,0,0 ; EXPECTED FLOAT RESULT
016230  000000
016232  047713  147704                          .WORD   047713,147704   ; FPS:  BEFORE, AFTER
```

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


                                ;;*****************************************************************
                                ;*TEST 233       TEST OF NEGD INSTR, DATA SET NEGD-12
                                ;*              ROUNDING MODE, ALL INTERRUPT ENABLES ON
                                ;;*****************************************************************
016240  000004                  TST233: SCOPE
016242  012705  016254                  MOV     #NEGD12,R5      ; PTR TO TEST DATA SET
016246  004737  022136                  JSR     PC,@#NEGDT      ; GO TEST

016252  000413                          BR      TST234          ;;

016254                          NEGD12: ; TEST DATA SET NEGD-12:
016254  100000  000000  000000          .WORD   M0,0,0,1        ; INITIAL MEM FLOAT NUMBER
016262  000001
016264  000000  000000  000000          .WORD   0,0,0,0 ; EXPECTED FLOAT RESULT
016272  000000
016274  047613  147604                  .WORD   047613,147604   ; FPS:  BEFORE, AFTER
016300  100014                          .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
                                ;;*****************************************************************
016302  000004                  TST234: SCOPE
016304  012705  016316                  MOV     #NEGD13,R5      ; PTR TO TEST DATA SET
016310  004737  022136                  JSR     PC,@#NEGDT      ; GO TEST

016314  000413                          BR      TST235          ;;

016316                          NEGD13: ; TEST DATA SET NEGD-13:
016316  100177  177777  177777          .WORD   ZX1MN,M1,M1,M1  ; INITIAL MEM FLOAT NUMBER
016324  177777
016326  000000  000000  000000          .WORD   0,0,0,0 ; EXPECTED FLOAT RESULT
016334  000000
016336  043753  043744                  .WORD   043753,043744   ; FPS:  BEFORE, AFTER
016342  000000                          .WORD   000000          ; FEC AFTER ( 0 = N/A )
```

```
                                ;;*****************************************************************
                                ;*TEST 235       TEST OF CLRF INSTR, DATA SET CLRF-1
                                ;*              TRUNCATE MODE, ALL INTERRUPT ENABLES ON
                                ;;*****************************************************************
016344  000004                  TST235: SCOPE
016346  012705  016360                  MOV     #CLRF1,R5       ; PTR TO TEST DATA SET
016352  004737  022346                  JSR     PC,@#CLRFT      ; GO TEST
016356  000407                          BR      TST236          ;;
016360                          CLRF1:  ; TEST DATA SET CLRF-1:
016360  000000  000000                  .WORD   0,0     ; INITIAL MEM FLOAT NUMBER
016364  000000  000000                  .WORD   0,0     ; EXPECTED FLOAT RESULT
016370  047553  047544                  .WORD   047553,047544   ; FPS:  BEFORE, AFTER
016374  000000                          .WORD   000000          ; FEC AFTER ( 0 = N/A )
                                ;;*****************************************************************
                                ;*TEST 236       TEST OF CLRF INSTR, DATA SET CLRF-2
                                ;*              ROUNDING MODE, ALL INTERRUPT ENABLES ON
                                ;;*****************************************************************
016376  000004                  TST236: SCOPE
016400  012705  016412                  MOV     #CLRF2,R5       ; PTR TO TEST DATA SET
016404  004737  022346                  JSR     PC,@#CLRFT      ; GO TEST
016410  000407                          BR      TST237          ;;
016412                          CLRF2:  ; TEST DATA SET CLRF-2:
016412  177777  177777                  .WORD   M1,M1   ; INITIAL MEM FLOAT NUMBER
016416  000000  000000                  .WORD   0,0     ; EXPECTED FLOAT RESULT
016422  047413  047404                  .WORD   047413,047404   ; FPS:  BEFORE, AFTER
016426  000000                          .WORD   000000          ; FEC AFTER ( 0 = N/A )
                                ;;*****************************************************************
                                ;*TEST 237       TEST OF CLRF INSTR, DATA SET CLRF-3
                                ;*              TRUNCATE MODE, ALL INTERRUPT ENABLES ON
                                ;;*****************************************************************
016430  000004                  TST237: SCOPE
016432  012705  016444                  MOV     #CLRF3,R5       ; PTR TO TEST DATA SET
016436  004737  022346                  JSR     PC,@#CLRFT      ; GO TEST
016442  000407                          BR      TST240          ;;
016444                          CLRF3:  ; TEST DATA SET CLRF-3:
016444  052525  052525                  .WORD   ALTP,ALTP       ; INITIAL MEM FLOAT NUMBER
016450  000000  000000                  .WORD   0,0     ; EXPECTED FLOAT RESULT
016454  047453  047444                  .WORD   047453,047444   ; FPS:  BEFORE, AFTER
016460  000000                          .WORD   000000          ; FEC AFTER ( 0 = N/A )
                                ;;*****************************************************************
                                ;*TEST 240       TEST OF CLRF INSTR, DATA SET CLRF-4
                                ;*              ROUNDING MODE, ALL INTERRUPT ENABLES ON
                                ;;*****************************************************************
016462  000004                  TST240: SCOPE
```

```
 4492  016464  012705  016476                  MOV     #CLRF4,R5       ; PTR TO TEST DATA SET
 4493  016470  004737  022346                  JSR     PC,@#CLRFT      ; GO TEST
 4494
 4495  016474  000407                          BR      TST241          ;;
 4496
 4497  016476                          CLRF4:  ; TEST DATA SET CLRF-4:
 4498  016476  125252  125252                  .WORD   ALTN,ALTN       ; INITIAL MEM FLOAT NUMBER
 4499  016502  000000  000000                  .WORD   0,0     ; EXPECTED FLOAT RESULT
 4500  016506  047513  047504                  .WORD   047513,047504   ; FPS:  BEFORE, AFTER
 4501  016512  000000                          .WORD   000000          ; FEC AFTER ( 0 = N/A )
 4502
 4503
 4504                                  ;;**********************************************************************
 4505                                  ;*TEST 241       TEST OF CLRF INSTR, DATA SET CLRF-5
 4506                                  ;*              TRUNCATE MODE, ALL INTERRUPT ENABLES ON
 4507                                  ;;**********************************************************************
 4508  016514  000004                  TST241: SCOPE
 4509  016516  012705  016530                  MOV     #CLRF5,R5       ; PTR TO TEST DATA SET
 4510  016522  004737  022346                  JSR     PC,@#CLRFT      ; GO TEST
 4511
 4512  016526  000407                          BR      TST242          ;;
 4513
 4514  016530                          CLRF5:  ; TEST DATA SET CLRF-5:
 4515  016530  100000  000000                  .WORD   M0,0    ; INITIAL MEM FLOAT NUMBER
 4516  016534  000000  000000                  .WORD   0,0     ; EXPECTED FLOAT RESULT
 4517  016540  047553  047544                  .WORD   047553,047544   ; FPS:  BEFORE, AFTER
 4518  016544  000000                          .WORD   000000          ; FEC AFTER ( 0 = N/A )
 4519
 4520
 4521                                  ;;**********************************************************************
 4522                                  ;*TEST 242       TEST OF CLRF INSTR, DATA SET CLRF-6
 4523                                  ;*              ROUNDING MODE, ALL INTERRUPT ENABLES ON
 4524                                  ;;**********************************************************************
 4525  016546  000004                  TST242: SCOPE
 4526  016550  012705  016562                  MOV     #CLRF6,R5       ; PTR TO TEST DATA SET
 4527  016554  004737  022346                  JSR     PC,@#CLRFT      ; GO TEST
 4528
 4529  016560  000407                          BR      TST243          ;;
 4530
 4531  016562                          CLRF6:  ; TEST DATA SET CLRF-6:
 4532  016562  000177  177777                  .WORD   ZX1MP,M1        ; INITIAL MEM FLOAT NUMBER
 4533  016566  000000  000000                  .WORD   0,0     ; EXPECTED FLOAT RESULT
 4534  016572  047413  047404                  .WORD   047413,047404   ; FPS:  BEFORE, AFTER
 4535  016576  000000                          .WORD   000000          ; FEC AFTER ( 0 = N/A )
 4536
 4537
 4538
```

```
                                        ;;*****************************************************************
                                        ;*TEST 243      TEST OF CLRD INSTR, DATA SET CLRD-1
                                        ;*             ROUNDING MODE, ALL INTERRUPT ENABLES ON
                                        ;;*****************************************************************
016600  000004                  TST243: SCOPE
016602  012705  016614                  MOV     #CLRD1,R5       ; PTR TO TEST DATA SET
016606  004737  022522                  JSR     PC,@#CLRDT      ; GO TEST

016612  000413                          BR      TST244          ;;

016614                          CLRD1:  ; TEST DATA SET CLRD-1:
016614  000000  000000  000000          .WORD   0,0,0,0 ; INITIAL MEM FLOAT NUMBER
016622  000000
016624  000000  000000  000000          .WORD   0,0,0,0 ; EXPECTED FLOAT RESULT
016632  000000
016634  047613  047604                  .WORD   047613,047604   ; FPS:  BEFORE, AFTER
016640  000000                          .WORD   000000          ; FEC AFTER ( 0 = N/A )


                                        ;;*****************************************************************
                                        ;*TEST 244      TEST OF CLRD INSTR, DATA SET CLRD-2
                                        ;*             TRUNCATE MODE, ALL INTERRUPT ENABLES ON
                                        ;;*****************************************************************
016642  000004                  TST244: SCOPE
016644  012705  016656                  MOV     #CLRD2,R5       ; PTR TO TEST DATA SET
016650  004737  022522                  JSR     PC,@#CLRDT      ; GO TEST

016654  000413                          BR      TST245          ;;

016656                          CLRD2:  ; TEST DATA SET CLRD-2:
016656  177777  177777  177777          .WORD   M1,M1,M1,M1     ; INITIAL MEM FLOAT NUMBER
016664  177777
016666  000000  000000  000000          .WORD   0,0,0,0 ; EXPECTED FLOAT RESULT
016674  000000
016676  047753  047744                  .WORD   047753,047744   ; FPS:  BEFORE, AFTER
016702  000000                          .WORD   000000          ; FEC AFTER ( 0 = N/A )


                                        ;;*****************************************************************
                                        ;*TEST 245      TEST OF CLRD INSTR, DATA SET CLRD-3
                                        ;*             TRUNCATE MODE, ALL INTERRUPT ENABLES ON
                                        ;;*****************************************************************
016704  000004                  TST245: SCOPE
016706  012705  016720                  MOV     #CLRD3,R5       ; PTR TO TEST DATA SET
016712  004737  022522                  JSR     PC,@#CLRDT      ; GO TEST

016716  000413                          BR      TST246          ;;

016720                          CLRD3:  ; TEST DATA SET CLRD-3:
016720  052525  052525  052525          .WORD   ALTP,ALTP,ALTP,ALTP     ; INITIAL MEM FLOAT NUMBER
016726  052525
016730  000000  000000  000000          .WORD   0,0,0,0 ; EXPECTED FLOAT RESULT
016736  000000
016740  047653  047644                  .WORD   047653,047644   ; FPS:  BEFORE, AFTER
016744  000000                          .WORD   000000          ; FEC AFTER ( 0 = N/A )
```

```
                                ;;*****************************************************************
                                ;*TEST 246       TEST OF CLRD INSTR, DATA SET CLRD-4
                                ;*              ROUNDING MODE, ALL INTERRUPT ENABLES ON
                                ;;*****************************************************************
016746  000004                  TST246: SCOPE
016750  012705  016762                  MOV     #CLRD4,R5       ; PTR TO TEST DATA SET
016754  004737  022522                  JSR     PC,@#CLRDT      ; GO TEST
016760  000413                          BR      TST247          ;;
016762                          CLRD4:  ; TEST DATA SET CLRD-4:
016762  125252  125252  125252          .WORD   ALTN,ALTN,ALTN,ALTN     ; INITIAL MEM FLOAT NUMBER
016770  125252
016772  000000  000000  000000          .WORD   0,0,0,0 ; EXPECTED FLOAT RESULT
017000  000000
017002  047713  047704                  .WORD   047713,047704   ; FPS:  BEFORE, AFTER
017006  000000                          .WORD   000000          ; FEC AFTER ( 0 = N/A )

                                ;;*****************************************************************
                                ;*TEST 247       TEST OF CLRD INSTR, DATA SET CLRD-5
                                ;*              ROUNDING MODE, ALL INTERRUPT ENABLES ON
                                ;;*****************************************************************
017010  000004                  TST247: SCOPE
017012  012705  017024                  MOV     #CLRD5,R5       ; PTR TO TEST DATA SET
017016  004737  022522                  JSR     PC,@#CLRDT      ; GO TEST
017022  000413                          BR      TST250          ;;
017024                          CLRD5:  ; TEST DATA SET CLRD-5:
017024  100000  000000  000000          .WORD   M0,0,0,0        ; INITIAL MEM FLOAT NUMBER
017032  000000
017034  000000  000000  000000          .WORD   0,0,0,0 ; EXPECTED FLOAT RESULT
017042  000000
017044  047613  047604                  .WORD   047613,047604   ; FPS:  BEFORE, AFTER
017050  000000                          .WORD   000000          ; FEC AFTER ( 0 = N/A )

                                ;;*****************************************************************
                                ;*TEST 250       TEST OF CLRD INSTR, DATA SET CLRD-6
                                ;*              TRUNCATE MODE, ALL INTERRUPT ENABLES ON
                                ;;*****************************************************************
017052  000004                  TST250: SCOPE
017054  012705  017066                  MOV     #CLRD6,R5       ; PTR TO TEST DATA SET
017060  004737  022522                  JSR     PC,@#CLRDT      ; GO TEST
017064  000413                          BR      TST251          ;;
017066                          CLRD6:  ; TEST DATA SET CLRD-6:
017066  000177  177777  177777          .WORD   ZX1MP,M1,M1,M1  ; INITIAL MEM FLOAT NUMBER
017074  177777
017076  000000  000000  000000          .WORD   0,0,0,0 ; EXPECTED FLOAT RESULT
017104  000000
017106  047753  047744                  .WORD   047753,047744   ; FPS:  BEFORE, AFTER
017112  000000                          .WORD   000000          ; FEC AFTER ( 0 = N/A )
```

```
4651
4652
4653
```

```
                                ;;*****************************************************************
                                ;*TEST 251       TEST OF TSTF INSTR, DATA SET TSTF-1
                                ;*              ROUNDING MODE, ALL INTERRUPT ENABLES ON
                                ;;*****************************************************************
017114  000004                  TST251: SCOPE
017116  012705  017130                  MOV     #TSTF1,R5       ; PTR TO TEST DATA SET
017122  004737  022732                  JSR     PC,@#TSTFT      ; GO TEST
                                
017126  000407                          BR      TST252          ;;
                                
017130                          TSTF1:  ; TEST DATA SET TSTF-1:
017130  000000  000000                  .WORD   0,0     ; INITIAL MEM FLOAT NUMBER
017134  000000  000000                  .WORD   0,0     ; EXPECTED FLOAT RESULT
017140  047513  047504                  .WORD   047513,047504   ; FPS:  BEFORE, AFTER
017144  000000                          .WORD   000000          ; FEC AFTER ( 0 = N/A )
                                
                                
                                ;;*****************************************************************
                                ;*TEST 252       TEST OF TSTF INSTR, DATA SET TSTF-2
                                ;*              TRUNCATE MODE, ALL INTERRUPT ENABLES ON
                                ;;*****************************************************************
017146  000004                  TST252: SCOPE
017150  012705  017162                  MOV     #TSTF2,R5       ; PTR TO TEST DATA SET
017154  004737  022732                  JSR     PC,@#TSTFT      ; GO TEST
                                
017160  000407                          BR      TST253          ;;
                                
017162                          TSTF2:  ; TEST DATA SET TSTF-2:
017162  052525  052525                  .WORD   ALTP,ALTP       ; INITIAL MEM FLOAT NUMBER
017166  052525  052525                  .WORD   ALTP,ALTP       ; EXPECTED FLOAT RESULT
017172  047557  047540                  .WORD   047557,047540   ; FPS:  BEFORE, AFTER
017176  000000                          .WORD   000000          ; FEC AFTER ( 0 = N/A )
                                
                                
                                ;;*****************************************************************
                                ;*TEST 253       TEST OF TSTF INSTR, DATA SET TSTF-3
                                ;*              ROUNDING MODE, ALL INTERRUPT ENABLES ON
                                ;;*****************************************************************
017200  000004                  TST253: SCOPE
017202  012705  017214                  MOV     #TSTF3,R5       ; PTR TO TEST DATA SET
017206  004737  022732                  JSR     PC,@#TSTFT      ; GO TEST
                                
017212  000407                          BR      TST254          ;;
                                
017214                          TSTF3:  ; TEST DATA SET TSTF-3:
017214  125252  125252                  .WORD   ALTN,ALTN       ; INITIAL MEM FLOAT NUMBER
017220  125252  125252                  .WORD   ALTN,ALTN       ; EXPECTED FLOAT RESULT
017224  047407  047410                  .WORD   047407,047410   ; FPS:  BEFORE, AFTER
017230  000000                          .WORD   000000          ; FEC AFTER ( 0 = N/A )
                                
                                
                                ;;*****************************************************************
                                ;*TEST 254       TEST OF TSTF INSTR, DATA SET TSTF-4
                                ;*              TRUNCATE MODE, ALL INTERRUPT ENABLES ON
                                ;;*****************************************************************
017232  000004                  TST254: SCOPE
```

```
                        012705  017246          MOV     #TSTF4,R5       ; PTR TO TEST DATA SET
                        004737  022732          JSR     PC,@#TSTFT      ; GO TEST

                        000407                  BR      TST255          ;;

                                        TSTF4:  ; TEST DATA SET TSTF-4:
                177777  177777                .WORD   LGN,M1  ; INITIAL MEM FLOAT NUMBER
                177777  177777                .WORD   LGN,M1  ; EXPECTED FLOAT RESULT
                047447  047450                .WORD   047447,047450   ; FPS:  BEFORE, AFTER
                000000                        .WORD   000000          ; FEC AFTER ( 0 = N/A )


                                        ;;**********************************************************************
                                        ;*TEST 255      TEST OF TSTF INSTR, DATA SET TSTF-5
                                        ;*             ROUNDING MODE, ALL INTERRUPT ENABLES ON
                                        ;;**********************************************************************
                000004                  TST255: SCOPE
                012705  017300                  MOV     #TSTF5,R5       ; PTR TO TEST DATA SET
                004737  022732                  JSR     PC,@#TSTFT      ; GO TEST

                000407                          BR      TST256          ;;

                                        TSTF5:  ; TEST DATA SET TSTF-5:
                000200  000000                .WORD   SMP,0   ; INITIAL MEM FLOAT NUMBER
                000200  000000                .WORD   SMP,0   ; EXPECTED FLOAT RESULT
                047517  047500                .WORD   047517,047500   ; FPS:  BEFORE, AFTER
                000000                        .WORD   000000          ; FEC AFTER ( 0 = N/A )


                                        ;;**********************************************************************
                                        ;*TEST 256      TEST OF TSTF INSTR, DATA SET TSTF-6
                                        ;*             TRUNCATE MODE, ALL INTERRUPT ENABLES ON
                                        ;;**********************************************************************
                000004                  TST256: SCOPE
                012705  017332                  MOV     #TSTF6,R5       ; PTR TO TEST DATA SET
                004737  022732                  JSR     PC,@#TSTFT      ; GO TEST

                000407                          BR      TST257          ;;

                                        TSTF6:  ; TEST DATA SET TSTF-6:
                077777  177777                .WORD   LGP,M1  ; INITIAL MEM FLOAT NUMBER
                077777  177777                .WORD   LGP,M1  ; EXPECTED FLOAT RESULT
                047557  047540                .WORD   047557,047540   ; FPS:  BEFORE, AFTER
                000000                        .WORD   000000          ; FEC AFTER ( 0 = N/A )


                                        ;;**********************************************************************
                                        ;*TEST 257      TEST OF TSTF INSTR, DATA SET TSTF-7
                                        ;*             ROUNDING MODE, ALL INTERRUPT ENABLES ON
                                        ;;**********************************************************************
                000004                  TST257: SCOPE
                012705  017364                  MOV     #TSTF7,R5       ; PTR TO TEST DATA SET
                004737  022732                  JSR     PC,@#TSTFT      ; GO TEST

                000407                          BR      TST260          ;;
```

```
FPU BASIC INSTR TESTS   MACY11 27(1006)  09-FEB-77  09:48  PAGE 100
DQFPAA.P11    09-FEB-77 09:42           T257    TEST OF TSTF INSTR, DATA SET TSTF-7

 4766  017364                          TSTF7:  ; TEST DATA SET TSTF-7:
 4767  017364  100200  000000                  .WORD   SMN,0   ; INITIAL MEM FLOAT NUMBER
 4768  017370  100200  000000                  .WORD   SMN,0   ; EXPECTED FLOAT RESULT
 4769  017374  047407  047410                  .WORD   047407,047410   ; FPS:  BEFORE, AFTER
 4770  017400  000000                          .WORD   000000          ; FEC AFTER ( 0 = N/A )
 4771
 4772
 4773                                  ;;*****************************************************************
 4774                                  ;*TEST 260      TEST OF TSTF INSTR, DATA SET TSTF-10
 4775                                  ;*             TRUNCATE MODE, ALL INTERRUPT ENABLES ON
 4776                                  ;;*****************************************************************
 4777  017402  000004                  TST260: SCOPE
 4778  017404  012705  017416                  MOV     #TSTF10,R5      ; PTR TO TEST DATA SET
 4779  017410  004737  022732                  JSR     PC,@#TSTFT      ; GO TEST
 4780
 4781  017414  000407                          BR      TST261          ;;
 4782
 4783  017416                          TSTF10: ; TEST DATA SET TSTF-10:
 4784  017416  000177  177777                  .WORD   ZX1MP,M1        ; INITIAL MEM FLOAT NUMBER
 4785  017422  000177  177777                  .WORD   ZX1MP,M1        ; EXPECTED FLOAT RESULT
 4786  017426  047453  047444                  .WORD   047453,047444   ; FPS:  BEFORE, AFTER
 4787  017432  000000                          .WORD   000000          ; FEC AFTER ( 0 = N/A )
 4788
 4789
 4790                                  ;;*****************************************************************
 4791                                  ;*TEST 261      TEST OF TSTF INSTR, DATA SET TSTF-11
 4792                                  ;*             ROUNDING MODE, ALL INTERRUPT ENABLES ON
 4793                                  ;;*****************************************************************
 4794  017434  000004                  TST261: SCOPE
 4795  017436  012705  017450                  MOV     #TSTF11,R5      ; PTR TO TEST DATA SET
 4796  017442  004737  022732                  JSR     PC,@#TSTFT      ; GO TEST
 4797
 4798  017446  000407                          BR      TST262          ;;
 4799
 4800  017450                          TSTF11: ; TEST DATA SET TSTF-11:
 4801  017450  100000  000000                  .WORD   M0,0    ; INITIAL MEM FLOAT NUMBER
 4802  017454  100000  000000                  .WORD   M0,0    ; EXPECTED FLOAT RESULT
 4803  017460  047503  147514                  .WORD   047503,147514   ; FPS:  BEFORE, AFTER
 4804  017464  100014                          .WORD   100014          ; FEC AFTER ( 0 = N/A )
 4805
 4806
 4807                                  ;;*****************************************************************
 4808                                  ;*TEST 262      TEST OF TSTF INSTR, DATA SET TSTF-12
 4809                                  ;*             TRUNCATE MODE, ALL INTERRUPT ENABLES ON
 4810                                  ;;*****************************************************************
 4811  017466  000004                  TST262: SCOPE
 4812  017470  012705  017502                  MOV     #TSTF12,R5      ; PTR TO TEST DATA SET
 4813  017474  004737  022732                  JSR     PC,@#TSTFT      ; GO TEST
 4814
 4815  017500  000407                          BR      TST263          ;;
 4816
 4817  017502                          TSTF12: ; TEST DATA SET TSTF-12:
 4818  017502  100000  000001                  .WORD   M0,1    ; INITIAL MEM FLOAT NUMBER
 4819  017506  100000  000001                  .WORD   M0,1    ; EXPECTED FLOAT RESULT
 4820  017512  047543  147554                  .WORD   047543,147554   ; FPS:  BEFORE, AFTER
 4821  017516  100014                          .WORD   100014          ; FEC AFTER ( 0 = N/A )
```

```
                                        ;;*****************************************************************
                                        ;*TEST 263      TEST OF TSTF INSTR, DATA SET TSTF-13
                                        ;*             ROUNDING MODE, -0 INTERRUPT ENABLE OFF, ALL OTHERS ON
                                        ;;*****************************************************************
017520  000004                          TST263: SCOPE
017522  012705  017534                          MOV     #TSTF13,R5      ; PTR TO TEST DATA SET
017526  004737  022732                          JSR     PC,@#TSTFT      ; GO TEST
017532  000407                                  BR      TST264          ;;
017534                                  TSTF13: ; TEST DATA SET TSTF-13:
017534  100177  177777                          .WORD   ZX1MN,M1        ; INITIAL MEM FLOAT NUMBER
017540  100177  177777                          .WORD   ZX1MN,M1        ; EXPECTED FLOAT RESULT
017544  043403  043414                          .WORD   043403,043414   ; FPS:  BEFORE, AFTER
017550  000000                                  .WORD   000000          ; FEC AFTER ( 0 = N/A )
```

```
                                ;;*****************************************************************
                                ;*TEST 264       TEST OF TSTD INSTR, DATA SET TSTD-1
                                ;*              TRUNCATE MODE, ALL INTERRUPT ENABLES ON
                                ;;*****************************************************************
017552  000004                  TST264: SCOPE
017554  012705  017566                  MOV     #TSTD1,R5       ; PTR TO TEST DATA SET
017560  004737  023106                  JSR     PC,@#TSTDT      ; GO TEST
                                
017564  000413                          BR      TST265          ;;
                                
017566                          TSTD1:  ; TEST DATA SET TSTD-1:
017566  000000  000000  000000          .WORD   0,0,0,0 ; INITIAL MEM FLOAT NUMBER
017574  000000
017576  000000  000000  000000          .WORD   0,0,0,0 ; EXPECTED FLOAT RESULT
017604  000000
017606  047653  047644                  .WORD   047653,047644   ; FPS:  BEFORE, AFTER
017612  000000                          .WORD   000000          ; FEC AFTER ( 0 = N/A )
                                
                                
                                ;;*****************************************************************
                                ;*TEST 265       TEST OF TSTD INSTR, DATA SET TSTD-2
                                ;*              ROUNDING MODE, ALL INTERRUPT ENABLES ON
                                ;;*****************************************************************
017614  000004                  TST265: SCOPE
017616  012705  017630                  MOV     #TSTD2,R5       ; PTR TO TEST DATA SET
017622  004737  023106                  JSR     PC,@#TSTDT      ; GO TEST
                                
017626  000413                          BR      TST266          ;;
                                
017630                          TSTD2:  ; TEST DATA SET TSTD-2:
017630  052525  052525  052525          .WORD   ALTP,ALTP,ALTP,ALTP     ; INITIAL MEM FLOAT NUMBER
017636  052525
017640  052525  052525  052525          .WORD   ALTP,ALTP,ALTP,ALTP     ; EXPECTED FLOAT RESULT
017646  052525
017650  047717  047700                  .WORD   047717,047700   ; FPS:  BEFORE, AFTER
017654  000000                          .WORD   000000          ; FEC AFTER ( 0 = N/A )
                                
                                
                                ;;*****************************************************************
                                ;*TEST 266       TEST OF TSTD INSTR, DATA SET TSTD-3
                                ;*              TRUNCATE MODE, ALL INTERRUPT ENABLES ON
                                ;;*****************************************************************
017656  000004                  TST266: SCOPE
017660  012705  017672                  MOV     #TSTD3,R5       ; PTR TO TEST DATA SET
017664  004737  023106                  JSR     PC,@#TSTDT      ; GO TEST
                                
017670  000413                          BR      TST267          ;;
                                
017672                          TSTD3:  ; TEST DATA SET TSTD-3:
017672  125252  125252  125252          .WORD   ALTN,ALTN,ALTN,ALTN     ; INITIAL MEM FLOAT NUMBER
017700  125252
017702  125252  125252  125252          .WORD   ALTN,ALTN,ALTN,ALTN     ; EXPECTED FLOAT RESULT
017710  125252
017712  047747  047750                  .WORD   047747,047750   ; FPS:  BEFORE, AFTER
017716  000000                          .WORD   000000          ; FEC AFTER ( 0 = N/A )
```

```
                                ;;*****************************************************************
                                ;*TEST 267       TEST OF TSTD INSTR, DATA SET TSTD-4
                                ;*              ROUNDING MODE, ALL INTERRUPT ENABLES ON
                                ;;*****************************************************************
017720  000004                  TST267: SCOPE
017722  012705  017734                  MOV     #TSTD4,R5       ; PTR TO TEST DATA SET
017726  004737  023106                  JSR     PC,@#TSTDT      ; GO TEST
                
017732  000413                          BR      TST270          ;;
                
017734                          TSTD4:  ; TEST DATA SET TSTD-4:
017734  177777  177777  177777          .WORD   LGN,M1,M1,M1    ; INITIAL MEM FLOAT NUMBER
017742  177777
017744  177777  177777  177777          .WORD   LGN,M1,M1,M1    ; EXPECTED FLOAT RESULT
017752  177777
017754  047607  047610                  .WORD   047607,047610   ; FPS:  BEFORE, AFTER
017760  000000                          .WORD   000000          ; FEC AFTER ( 0 = N/A )
                
                                ;;*****************************************************************
                                ;*TEST 270       TEST OF TSTD INSTR, DATA SET TSTD-5
                                ;*              TRUNCATE MODE, ALL INTERRUPT ENABLES ON
                                ;;*****************************************************************
017762  000004                  TST270: SCOPE
017764  012705  017776                  MOV     #TSTD5,R5       ; PTR TO TEST DATA SET
017770  004737  023106                  JSR     PC,@#TSTDT      ; GO TEST
                
017774  000413                          BR      TST271          ;;
                
017776                          TSTD5:  ; TEST DATA SET TSTD-5:
017776  000200  000000  000000          .WORD   SMP,0,0,0       ; INITIAL MEM FLOAT NUMBER
020004  000000
020006  000200  000000  000000          .WORD   SMP,0,0,0       ; EXPECTED FLOAT RESULT
020014  000000
020016  047657  047640                  .WORD   047657,047640   ; FPS:  BEFORE, AFTER
020022  000000                          .WORD   000000          ; FEC AFTER ( 0 = N/A )
                
                                ;;*****************************************************************
                                ;*TEST 271       TEST OF TSTD INSTR, DATA SET TSTD-6
                                ;*              ROUNDING MODE, ALL INTERRUPT ENABLES ON
                                ;;*****************************************************************
020024  000004                  TST271: SCOPE
020026  012705  020040                  MOV     #TSTD6,R5       ; PTR TO TEST DATA SET
020032  004737  023106                  JSR     PC,@#TSTDT      ; GO TEST
                
020036  000413                          BR      TST272          ;;
                
020040                          TSTD6:  ; TEST DATA SET TSTD-6:
020040  077777  177777  177777          .WORD   LGP,M1,M1,M1    ; INITIAL MEM FLOAT NUMBER
020046  177777
020050  077777  177777  177777          .WORD   LGP,M1,M1,M1    ; EXPECTED FLOAT RESULT
020056  177777
020060  047717  047700                  .WORD   047717,047700   ; FPS:  BEFORE, AFTER
020064  000000                          .WORD   000000          ; FEC AFTER ( 0 = N/A )
```

```
                                ;;*****************************************************************
                                ;*TEST 272       TEST OF TSTD INSTR, DATA SET TSTD-7
                                ;*              TRUNCATE MODE, ALL INTERRUPT ENABLES ON
                                ;;*****************************************************************
020066  000004                  TST272: SCOPE
020070  012705  020102                  MOV     #TSTD7,R5       ; PTR TO TEST DATA SET
020074  004737  023106                  JSR     PC,@#TSTDT      ; GO TEST
020100  000413                          BR      TST273          ;;
020102                          TSTD7:  ; TEST DATA SET TSTD-7:
020102  100200  000000  000000          .WORD   SMN,0,0,0       ; INITIAL MEM FLOAT NUMBER
020110  000000
020112  100200  000000  000000          .WORD   SMN,0,0,0       ; EXPECTED FLOAT RESULT
020120  000000
020122  047747  047750                  .WORD   047747,047750   ; FPS:  BEFORE, AFTER
020126  000000                          .WORD   000000          ; FEC AFTER ( 0 = N/A )

                                ;;*****************************************************************
                                ;*TEST 273       TEST OF TSTD INSTR, DATA SET TSTD-10
                                ;*              ROUNDING MODE, ALL INTERRUPT ENABLES ON
                                ;;*****************************************************************
020130  000004                  TST273: SCOPE
020132  012705  020144                  MOV     #TSTD10,R5      ; PTR TO TEST DATA SET
020136  004737  023106                  JSR     PC,@#TSTDT      ; GO TEST
020142  000413                          BR      TST274          ;;
020144                          TSTD10: ; TEST DATA SET TSTD-10:
020144  000177  177777  177777          .WORD   ZX1MP,M1,M1,M1  ; INITIAL MEM FLOAT NUMBER
020152  177777
020154  000177  177777  177777          .WORD   ZX1MP,M1,M1,M1  ; EXPECTED FLOAT RESULT
020162  177777
020164  047613  047604                  .WORD   047613,047604   ; FPS:  BEFORE, AFTER
020170  000000                          .WORD   000000          ; FEC AFTER ( 0 = N/A )

                                ;;*****************************************************************
                                ;*TEST 274       TEST OF TSTD INSTR, DATA SET TSTD-11
                                ;*              TRUNCATE MODE, ALL INTERRUPT ENABLES ON
                                ;;*****************************************************************
020172  000004                  TST274: SCOPE
020174  012705  020206                  MOV     #TSTD11,R5      ; PTR TO TEST DATA SET
020200  004737  023106                  JSR     PC,@#TSTDT      ; GO TEST
020204  000413                          BR      TST275          ;;
020206                          TSTD11: ; TEST DATA SET TSTD-11:
020206  100000  000000  000000          .WORD   M0,0,0,0        ; INITIAL MEM FLOAT NUMBER
020214  000000
020216  100000  000000  000000          .WORD   M0,0,0,0        ; EXPECTED FLOAT RESULT
020224  000000
020226  047643  147654                  .WORD   047643,147654   ; FPS:  BEFORE, AFTER
```

```
5010  020232  100014                          .WORD   100014          ; FEC AFTER ( 0 = N/A )
5011
5012
5013                                  ;;*****************************************************************
5014                                  ;*TEST 275       TEST OF TSTD INSTR, DATA SET TSTD-12
5015                                  ;*              ROUNDING MODE, ALL INTERRUPT ENABLES ON
5016                                  ;;*****************************************************************
5017  020234  000004                  TST275: SCOPE
5018  020236  012705  020250                  MOV     #TSTD12,R5      ; PTR TO TEST DATA SET
5019  020242  004737  023106                  JSR     PC,@#TSTDT      ; GO TEST
5020
5021  020246  000413                          BR      TST276          ;;
5022
5023  020250                          TSTD12: ; TEST DATA SET TSTD-12:
5024  020250  100000  000000  000000          .WORD   M0,0,0,1        ; INITIAL MEM FLOAT NUMBER
5025  020256  000001
5026  020260  100000  000000  000000          .WORD   M0,0,0,1        ; EXPECTED FLOAT RESULT
5027  020266  000001
5028  020270  047703  147714                  .WORD   047703,147714   ; FPS:  BEFORE, AFTER
5029  020274  100014                          .WORD   100014          ; FEC AFTER ( 0 = N/A )
5030
5031
5032                                  ;;*****************************************************************
5033                                  ;*TEST 276       TEST OF TSTD INSTR, DATA SET TSTD-13
5034                                  ;*              TRUNCATE MODE, -0 INTERRUPT ENABLE OFF, ALL OTHERS ON
5035                                  ;;*****************************************************************
5036  020276  000004                  TST276: SCOPE
5037  020300  012705  020312                  MOV     #TSTD13,R5      ; PTR TO TEST DATA SET
5038  020304  004737  023106                  JSR     PC,@#TSTDT      ; GO TEST
5039
5040  020310  000413                          BR      TST277          ;;
5041
5042  020312                          TSTD13: ; TEST DATA SET TSTD-13:
5043  020312  100177  177777  177777          .WORD   ZX1MN,M1,M1,M1  ; INITIAL MEM FLOAT NUMBER
5044  020320  177777
5045  020322  100177  177777  177777          .WORD   ZX1MN,M1,M1,M1  ; EXPECTED FLOAT RESULT
5046  020330  177777
5047  020332  043643  043654                  .WORD   043643,043654   ; FPS:  BEFORE, AFTER
5048  020336  000000                          .WORD   000000          ; FEC AFTER ( 0 = N/A )
5049
5050
5051
```

```
                                ;;*****************************************************************
                                .SBTTL  SUB PASS END CONTROL

020340                          TST277: ;FORCE LAST TEST NUMBER
020340  000004                          $SCOPE          ;CHECK FOR TEST ITERATIONS HERE

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

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

020342  076600  000022                  MED     @RWHAMI         ;GET WHAMI INTO R0
020346  032700  000020                  BIT     #BIT04,R0       ;1=HFP PRESENT, 0=NONE
020352  001423                          BEQ     $EOP            ;EXIT IF NONE

020354  032777  000002  160564          BIT     #SW01,@SWR      ;1=HFP OR WFP TEST ONLY
020362  001017                          BNE     $EOP            ;0=ALTERNATE HFP AND WFP TESTS

020364  012701  010000                  MOV     #BIT12,R1       ;HFP PRESENT, AND IN ALTERNATE MODE;
020370  076600  000144                  MED     @RFLAG          ;SO READ FLAGS,
020374  030100                          BIT     R1,R0           ;COMPLEMENT FLAG(5)=BIT12=HFP ENABLE FLAG
020376  001402                          BEQ     1$              ;
020400  040100                          BIC     R1,R0           ;CLEAR BIT 12
020402  000401                          BR      2$              ;
020404  050100                  1$:     BIS     R1,R0           ;SET BIT 12
020406  076600  000344          2$:     MED     @WFLAG          ;REWRITE FLAGS

020412  030100                          BIT     R1,R0           ;HFP OR WFP NEXT ?
020414  001002                          BNE     $EOP            ;IF HFP AGAIN, START NEW PASS
020416  000137  003200                  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


020422                          $EOP:
020422  005037  001104                  CLR     $ERFLG          ;ZERO ERROR COUNT
020426  005037  001102                  CLR     $TSTNM          ;ZERO TEST NUMBER
020432  005037  001310                  CLR     $TIMES          ;ZERO NUMBER OF ITERATIONS
020436  005237  001332                  INC     $PASS           ;INCREMENT PASS COUNT,
020442  042737  100000  001332          BIC     #100000,$PASS   ;  BUT NEVER LET IN GO NEGATIVE
020450  005327                          DEC     (PC)+           ;PASS LOOP ?
020452  000001                  $EOPCT: .WORD   1
020454  003021                          BGT     $DOAGN          ;YES
```

```
5108  020456  012737                          MOV     (PC)+,@(PC)+    ;RESTORE COUNTER
5109  020460  000001              $ENDCT:     .WORD   1               ;
5110  020462  020452                          $EOPCT                  ;
5111  020464  032777  002000  160454          BIT     #SW10,@SWR      ;BELL ON PASS END ?
5112  020472  001002                          BNE     $GET42          ;NO
5113  020474  104401  001314                  TYPE    ,$BELL          ;YES
5114
5115  020500  013700  000042          $GET42: MOV     @#42,R0         ;GET MONITOR ADDRESS
5116  020504  001405                          BEQ     $DOAGN          ;NO MONITOR
5117  020506  000005                          RESET                   ;CLEAR WORLD
5118
5119  020510  004710                  $ENDAD: JSR     PC,(R0)         ;GO TO MONITOR
5120  020512  000240                          NOP                     ;
5121  020514  000240                          NOP                     ;RESERVED FOR ACT11
5122  020516  000240                          NOP                     ;
5123
5124  020520  000137  003142          $DOAGN: JMP     @#NEWPAS        ;RETURN
5125
5126
```

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

020524                                  LDADT:
020524  012700  000013                          MOV     #13,R0          ; LOAD $TMP0-12
020530  010501                                  MOV     R5,R1           ;   WITH TEST DATA SETS
020532  012702  001230                          MOV     #$TMP0,R2       ;   FOR DISPLAY LATER
020536  012122                                  MOV     (R1)+,(R2)+     ;
020540  077002                                  SOB     R0,.-2          ;
020542  012737  020550  001114                  MOV     #LDADL,$LPERR   ; ERROR LOOPING ADDRESS

020550  170165  000020          LDADL:  LDFPS   20(R5)          ; INITIAL FPS

020554  004714                          JSR     PC,(R4)         ; GO PERFORM LDD/STD SEQUENCE

020556  170237  002330                  STFPS   FPS             ; STORE FPS AFTER
020562  170337  002332                  STST    FEC             ; STORE FEC/FEA AFTER

020566  023765  002330  000022          CMP     FPS,22(R5)      ; CHECK FPS
020574  001401                          BEQ     65$             ; FPS IS OK
020576  104077                          ERROR   77              ; FPS BAD
020600  005765  000124          65$:    TST     24(R5)          ; DOES FEC/FEA APPLY?
020604  100016                          BPL     66$             ; NO - SKIP TEST
020606  010437  002344                  MOV     R4,EXPFEA       ; GET EXPECTED FEA
020612  062737  000004  002344          ADD     #4,EXPFEA       ;   AND ADJUST
020620  123765  002332  000024          CMPB    FEC,24(R5)      ; COMPARE FEC-S
020626  001004                          BNE     64$             ; NOT EQUAL
020630  023737  002334  002344          CMP     FEA,EXPFEA      ; COMPARE FEA-S
020636  001401                          BEQ     66$             ; FEC, FEA OK
020640  104102                  64$:    ERROR   102             ; FEC OR FEA ARE BAD
020642                          66$:

020642  023765  001170  000010          CMP     $REG0,10(R5)    ; 1ST WORD OF RESULT CHECK?
020650  001014                          BNE     67$             ; NO
020652  023765  001172  000012          CMP     $REG1,12(R5)    ; 2ND WORD OF RESULT CHECK?
020660  001010                          BNE     67$             ; NO
020662  023765  001174  000014          CMP     $REG2,14(R5)    ; 3RD WORD OF RESULT CHECK?
020670  001004                          BNE     67$             ; NO
020672  023765  001176  000016          CMP     $REG3,16(R5)    ; 4TH WORD OF RESULT CHECK?
020700  001401                          BEQ     68$             ; ALL WORDS OK
020702  104104                  67$:    ERROR   104             ; NUMBERS NOT EQUAL
020704                          68$:

020704  000207                          RTS     PC              ; RETURN TO TEST CALLER

                                        ; * * *  LDD/STD SEQUENCE SUBR  * * * * * * * * * * * * * * * * * *

020706  172437  002366          LDARD0: LDD     PREVAC,AC0      ; PREV AC0 CONTENTS
020712  172415                          LDD     (R5),AC0        ; LOAD AC0
020714  174037  001170                  STD     AC0,$REG0       ; STORE AC0
020720  000207                          RTS     PC              ;
020722  172537  002366          LDARD1: LDD     PREVAC,AC1      ; PREV AC1 CONTENTS
020726  172515                          LDD     (R5),AC1        ; LOAD AC1
020730  174137  001170                  STD     AC1,$REG0       ; STORE AC1
020734  000207                          RTS     PC              ;
020736  172637  002366          LDARD2: LDD     PREVAC,AC2      ; PREV AC2 CONTENTS
020742  172615                          LDD     (R5),AC2        ; LOAD AC2
```

```
 5183  020744  174237  001170                  STD     AC2,$REGO       ; STORE AC2
 5184  020750  000207                          RTS     PC              ;
 5185  020752  172737  002366          LDARD3: LDD     PREVAC,AC3      ; PREV AC3 CONTENTS
 5186  020756  172715                          LDD     (R5),AC3        ; LOAD AC3
 5187  020760  174337  001170                  STD     AC3,$REGO       ; STORE AC3
 5188  020764  000207                          RTS     PC              ;
 5189  020766  172437  002366          LDARD4: LDD     PREVAC,AC0      ; PREV AC0 CONTENTS
 5190  020772  172415                          LDD     (R5),AC0        ; INTERMEDIATE
 5191  020774  174004                          STD     AC0,AC4         ;   LOAD AC4
 5192  020776  172504                          LDD     AC4,AC1         ; INTERMEDIATE
 5193  021000  174137  001170                  STD     AC1,$REGO       ;   STORE AC4
 5194  021004  000207                          RTS     PC              ;
 5195  021006  172537  002366          LDARD5: LDD     PREVAC,AC1      ; PREV AC1 CONTENTS
 5196  021012  172515                          LDD     (R5),AC1        ; INTERMEDIATE
 5197  021014  174105                          STD     AC1,AC5         ;   LOAD AC5
 5198  021016  172405                          LDD     AC5,AC0         ; INTERMEDIATE
 5199  021020  174037  001170                  STD     AC0,$REGO       ;   STORE AC5
 5200  021024  000207                          RTS     PC              ;
 5201
```

```
                                        .SBTTL  SUBR TO PERFORM TEST OF LDD/LDF/STD
                                
021026                          LDAFT:
021026  012700  000015                  MOV     #15,R0          ; LOAD $TMP0-14
021032  010501                          MOV     R5,R1           ;   WITH TEST DATA SETS
021034  012702  001230                  MOV     #$TMP0,R2       ;   FOR DISPLAY LATER
021040  012122                          MOV     (R1)+,(R2)+     ;
021042  077002                          SOB     R0,.-2          ;
021044  012737  021052  001114          MOV     #LDAFL,$LPERR   ; ERROR LOOPING ADDRESS
                                
021052  170011                  LDAFL:  SETD                    ; D MODE FOR INITIAL LOAD
                                
021054  004714                          JSR     PC,(R4)         ; GO PERFORM LDD/LDF/STD SEQUENCE
                                
021056  170237  002330                  STFPS   FPS             ; STORE FPS AFTER
021062  170337  002332                  STST    FEC             ; STORE FEC/FEA AFTER
                                
021066  023765  002330  000026          CMP     FPS,26(R5)      ; CHECK FPS
021074  001401                          BEQ     65$             ; FPS IS OK
021076  104100                          ERROR   100             ; FPS BAD
021100  005765  000030          65$:    TST     30(R5)          ; DOES FEC/FEA APPLY?
021104  100016                          BPL     66$             ; NO - SKIP TEST
021106  010437  002344                  MOV     R4,EXPFEA       ; GET EXPECTED FEA
021112  062737  000006  002344          ADD     #6,EXPFEA       ;   AND ADJUST
021120  123765  002332  000030          CMPB    FEC,30(R5)      ; COMPARE FEC-S
021126  001004                          BNE     64$             ; NOT EQUAL
021130  023737  002334  002344          CMP     FEA,EXPFEA      ; COMPARE FEA-S
021136  001401                          BEQ     66$             ; FEC, FEA OK
021140  104103                  64$:    ERROR   103             ; FEC OR FEA ARE BAD
021142                          66$:
                                
021142  023765  001170  000014          CMP     $REG0,14(R5)    ; 1ST WORD OF RESULT CHECK?
021150  001014                          BNE     67$             ; NO
021152  023765  001172  000016          CMP     $REG1,16(R5)    ; 2ND WORD OF RESULT CHECK?
021160  001010                          BNE     67$             ; NO
021162  023765  001174  000020          CMP     $REG2,20(R5)    ; 3RD WORD OF RESULT CHECK?
021170  001004                          BNE     67$             ; NO
021172  023765  001176  000022          CMP     $REG3,22(R5)    ; 4TH WORD OF RESULT CHECK?
021200  001401                          BEQ     68$             ; ALL WORDS OK
021202  104105                  67$:    ERROR   105             ; NUMBERS NOT EQUAL
021204                          68$:
                                
021204  000207                          RTS     PC              ; RETURN TO TEST CALLER
                                
                                ; * * *  LDD/LDF/STD SEQUENCE SUBR  * * * * * * * * * * * * * * * * * *
                                
021206  172415                  LDARF0: LDD     (R5),AC0        ; PREV AC0 CONTENTS
021210  170165  000024                  LDFPS   24(R5)          ; INITIAL FPS
021214  172465  000010                  LDF     10(R5),AC0      ; LOAD NEW HALF OF AC0
021220  170011                          SETD                    ; D MODE FOR STORE
021222  174037  001170                  STD     AC0,$REG0       ; STORE ALL OF RESULT
021226  000207                          RTS     PC              ;
021230  172515                  LDARF1: LDD     (R5),AC1        ; PREV AC1 CONTENTS
021232  170165  000024                  LDFPS   24(R5)          ; INITIAL FPS
021236  172565  000010                  LDF     10(R5),AC1      ; LOAD NEW HALF OF AC1
021242  170011                          SETD                    ; D MODE FOR STORE
```

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

```
                                        .SBTTL  SUBR TO TEST THE ABSF INSTRUCTION
                                        ABSFT:
021376                                          MOV     #7,R0           ; LOAD $TMP0-6
021376  012700  000007                          MOV     R5,R1           ;   WITH TEST DATA SETS
021402  010501                                  MOV     #$TMP0,R2       ;   FOR DISPLAY LATER
021404  012702  001230                          MOV     (R1)+,(R2)+     ;
021410  012122                                  SOB     R0,.-2          ;
021412  077002                                  MOV     #ABSFL,$LPERR   ; ERROR LOOPING ADDRESS
021414  012737  021422  001114
                                        ABSFL:  SETF                    ; F MODE
021422  170001                                  MOV     (R5),$REG0      ; INITIAL FLOAT NUMBER
021424  011537  001170                          MOV     2(R5),$REG1     ;   INTO OPERATING ROOM
021430  016537  000002  001172                  LDFPS   10(R5)          ; INITIAL FPS
021436  170165  000010
                                        ABSFI:  ABSF    $REG0           ; ABS[($REG0)]->$REG0
021442  170637  001170
                                                STFPS   FPS             ; STORE FPS AFTER
021446  170237  002330                          STST    FEC             ; STORE FEC/FEA AFTER
021452  170337  002332
                                                CMP     FPS,12(R5)      ; CHECK FPS
021456  023765  002330  000012                  BEQ     65$             ; FPS IS OK
021464  001401                                  ERROR   76              ; FPS BAD
021466  104076                          65$:    TST     14(R5)          ; DOES FEC/FEA APPLY?
021470  005765  000014                          BPL     66$             ; NO - SKIP TEST
021474  100014                                  MOV     #ABSFI,EXPFEA   ; GET EXPECTED FEA
021476  012737  021442  002344                  CMPB    FEC,14(R5)      ; COMPARE FEC-S
021504  123765  002332  000014                  BNE     64$             ; NOT EQUAL
021512  001004                                  CMP     FEA,EXPFEA      ; COMPARE FEA-S
021514  023737  002334  002344                  BEQ     66$             ; FEC, FEA OK
021522  001401                          64$:    ERROR   101             ; FEC OR FEA ARE BAD
021524  104101                          66$:
021526
                                                CMP     $REG0,4(R5)     ; 1ST WORD OF RESULT CHECK?
021526  023765  001170  000004                  BNE     67$             ; NO
021534  001004                                  CMP     $REG1,6(R5)     ; 2ND WORD OF RESULT CHECK?
021536  023765  001172  000006                  BEQ     68$             ; ALL WORDS OK
021544  001401                          67$:    ERROR   106             ; NUMBERS NOT EQUAL
021546  104106                          68$:
021550
                                                RTS     PC              ; RETURN TO TEST CALLER
021550  000207
                                        ;;*****************************************************************
                                        .SBTTL  SUBR TO TEST THE ABSD INSTRUCTION
                                        ABSDT:
021552                                          MOV     #13,R0          ; LOAD $TMP0-12
021552  012700  000013                          MOV     R5,R1           ;   WITH TEST DATA SETS
021556  010501                                  MOV     #$TMP0,R2       ;   FOR DISPLAY LATER
021560  012702  001230                          MOV     (R1)+,(R2)+     ;
021564  012122                                  SOB     R0,.-2          ;
021566  077002                                  MOV     #ABSDL,$LPERR   ; ERROR LOOPING ADDRESS
021570  012737  021576  001114
                                        ABSDL:  SETD                    ; D MODE
021576  170011                                  MOV     (R5),$REG0      ; INITIAL FLOAT NUMBER
021600  011537  001170                          MOV     2(R5),$REG1     ;   INTO OPERATING ROOM
021604  016537  000002  001172
```

```
5346  021612  016537  000004  001174          MOV     4(R5),$REG2     ;
5347  021620  016537  000006  001176          MOV     6(R5),$REG3     ;
5348  021626  170165  000020                  LDFPS   20(R5)          ; INITIAL FPS
5349
5350  021632  170637  001170          ABSDI:  ABSD    $REG0           ; ABS[($REG0)]->$REG0
5351
5352  021636  170237  002330                  STFPS   FPS             ; STORE FPS AFTER
5353  021642  170337  002332                  STST    FEC             ; STORE FEC/FEA AFTER
5354
5355  021646  023765  002330  000022          CMP     FPS,22(R5)      ; CHECK FPS
5356  021654  001401                          BEQ     65$             ; FPS IS OK
5357  021656  104077                          ERROR   77              ; FPS BAD
5358  021660  005765  000024          65$:    TST     24(R5)          ; DOES FEC/FEA APPLY?
5359  021664  100014                          BPL     66$             ; NO - SKIP TEST
5360  021666  012737  021632  002344          MOV     #ABSDI,EXPFEA   ; GET EXPECTED FEA
5361  021674  123765  002332  000024          CMPB    FEC,24(R5)      ; COMPARE FEC-S
5362  021702  001004                          BNE     64$             ; NOT EQUAL
5363  021704  023737  002334  002344          CMP     FEA,EXPFEA      ; COMPARE FEA-S
5364  021712  001401                          BEQ     66$             ; FEC, FEA OK
5365  021714  104102                  64$:    ERROR   102             ; FEC OR FEA ARE BAD
5366  021716                          66$:
5367
5368  021716  023765  001170  000010          CMP     $REG0,10(R5)    ; 1ST WORD OF RESULT CHECK?
5369  021724  001014                          BNE     67$             ; NO
5370  021726  023765  001172  000012          CMP     $REG1,12(R5)    ; 2ND WORD OF RESULT CHECK?
5371  021734  001010                          BNE     67$             ; NO
5372  021736  023765  001174  000014          CMP     $REG2,14(R5)    ; 3RD WORD OF RESULT CHECK?
5373  021744  001004                          BNE     67$             ; NO
5374  021746  023765  001176  000016          CMP     $REG3,16(R5)    ; 4TH WORD OF RESULT CHECK?
5375  021754  001401                          BEQ     68$             ; ALL WORDS OK
5376  021756  104107                  67$:    ERROR   107             ; NUMBERS NOT EQUAL
5377  021760                          68$:
5378
5379  021760  000207                          RTS     PC              ; RETURN TO TEST CALLER
5380
```

```
5381                                        .SBTTL  SUBR TO TEST THE NEGF INSTRUCTION
5382
5383  021762                                NEGFT:
5384  021762  012700  000007                        MOV     #7,R0           ; LOAD $TMP0-6
5385  021766  010501                              MOV     R5,R1           ;   WITH TEST DATA SETS
5386  021770  012702  001230                      MOV     #$TMP0,R2       ;   FOR DISPLAY LATER
5387  021774  012122                              MOV     (R1)+,(R2)+     ;
5388  021776  077002                              SOB     R0,.-2          ;
5389  022000  012737  022006  001114              MOV     #NEGFL,$LPERR   ; ERROR LOOPING ADDRESS
5390
5391  022006  170001                      NEGFL:  SETF                    ; F MODE
5392  022010  011537  001170                      MOV     (R5),$REG0      ; INITIAL FLOAT NUMBER
5393  022014  016537  000002  001172              MOV     2(R5),$REG1     ;   INTO OPERATING ROOM
5394  022022  170165  000010                      LDFPS   10(R5)          ; INITIAL FPS
5395
5396  022026  170737  001170              NEGFI:  NEGF    $REG0           ; -($REG0)->$REG0
5397
5398  022032  170237  002330                      STFPS   FPS             ; STORE FPS AFTER
5399  022036  170337  002332                      STST    FEC             ; STORE FEC/FEA AFTER
5400
5401  022042  023765  002330  000012              CMP     FPS,12(R5)      ; CHECK FPS
5402  022050  001401                              BEQ     65$             ; FPS IS OK
5403  022052  104076                              ERROR   76              ; FPS BAD
5404  022054  005765  000014                65$:  TST     14(R5)          ; DOES FEC/FEA APPLY?
5405  022060  100014                              BPL     66$             ; NO - SKIP TEST
5406  022062  012737  022026  002344              MOV     #NEGFI,EXPFEA   ; GET EXPECTED FEA
5407  022070  123765  002332  000014              CMPB    FEC,14(R5)      ; COMPARE FEC-S
5408  022076  001004                              BNE     64$             ; NOT EQUAL
5409  022100  023737  002334  002344              CMP     FEA,EXPFEA      ; COMPARE FEA-S
5410  022106  001401                              BEQ     66$             ; FEC, FEA OK
5411  022110  104101                        64$:  ERROR   101             ; FEC OR FEA ARE BAD
5412  022112                                66$:
5413
5414  022112  023765  001170  000004              CMP     $REG0,4(R5)     ; 1ST WORD OF RESULT CHECK?
5415  022120  001004                              BNE     67$             ; NO
5416  022122  023765  001172  000006              CMP     $REG1,6(R5)     ; 2ND WORD OF RESULT CHECK?
5417  022130  001401                              BEQ     68$             ; ALL WORDS OK
5418  022132  104110                        67$:  ERROR   110             ; NUMBERS NOT EQUAL
5419  022134                                68$:
5420
5421  022134  000207                              RTS     PC              ; RETURN TO TEST CALLER
5422
5423                                        ;;*****************************************************************
5424                                        .SBTTL  SUBR TO TEST THE NEGD INSTRUCTION
5425
5426  022136                                NEGDT:
5427  022136  012700  000013                      MOV     #13,R0          ; LOAD $TMP0-12
5428  022142  010501                              MOV     R5,R1           ;   WITH TEST DATA SETS
5429  022144  012702  001230                      MOV     #$TMP0,R2       ;   FOR DISPLAY LATER
5430  022150  012122                              MOV     (R1)+,(R2)+     ;
5431  022152  077002                              SOB     R0,.-2          ;
5432  022154  012737  022162  001114              MOV     #NEGDL,$LPERR   ; ERROR LOOPING ADDRESS
5433
5434  022162  170011                      NEGDL:  SETD                    ; D MODE
5435  022164  011537  001170                      MOV     (R5),$REG0      ; INITIAL FLOAT NUMBER
5436  022170  016537  000002  001172              MOV     2(R5),$REG1     ;   INTO OPERATING ROOM
```

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

```
                                        .SBTTL  SUBR TO TEST THE CLRF INSTRUCTION
                                        CLRFT:
022346                                          MOV     #7,R0           ; LOAD $TMP0-6
022346  012700  000007                          MOV     R5,R1           ;   WITH TEST DATA SETS
022352  010501                                  MOV     #$TMP0,R2       ;   FOR DISPLAY LATER
022354  012702  001230                          MOV     (R1)+,(R2)+     ;
022360  012122                                  SOB     R0,.-2          ;
022362  077002                                  MOV     #CLRFL,$LPERR   ; ERROR LOOPING ADDRESS
022364  012737  022372  001114
                                        CLRFL:  SETF                    ; F MODE
022372  170001                                  MOV     (R5),$REG0      ; INITIAL FLOAT NUMBER
022374  011537  001170                          MOV     2(R5),$REG1     ;   INTO OPERATING ROOM
022400  016537  000002  001172                  LDFPS   10(R5)          ; INITIAL FPS
022406  170165  000010
                                        CLRFI:  CLRF    $REG0           ; 0->$REG0
022412  170437  001170
022416  170237  002330                          STFPS   FPS             ; STORE FPS AFTER
022422  170337  002332                          STST    FEC             ; STORE FEC/FEA AFTER
022426  023765  002330  000012                  CMP     FPS,12(R5)      ; CHECK FPS
022434  001401                                  BEQ     65$             ; FPS IS OK
022436  104076                                  ERROR   76              ; FPS BAD
022440  005765  000014                  65$:    TST     14(R5)          ; DOES FEC/FEA APPLY?
022444  100014                                  BPL     66$             ; NO - SKIP TEST
022446  012737  022412  002344                  MOV     #CLRFI,EXPFEA   ; GET EXPECTED FEA
022454  123765  002332  000014                  CMPB    FEC,14(R5)      ; COMPARE FEC-S
022462  001004                                  BNE     64$             ; NOT EQUAL
022464  023737  002334  002344                  CMP     FEA,EXPFEA      ; COMPARE FEA-S
022472  001401                                  BEQ     66$             ; FEC, FEA OK
022474  104101                          64$:    ERROR   101             ; FEC OR FEA ARE BAD
022476                                  66$:
022476  023765  001170  000004                  CMP     $REG0,4(R5)     ; 1ST WORD OF RESULT CHECK?
022504  001004                                  BNE     67$             ; NO
022506  023765  001172  000006                  CMP     $REG1,6(R5)     ; 2ND WORD OF RESULT CHECK?
022514  001401                                  BEQ     68$             ; ALL WORDS OK
022516  104112                          67$:    ERROR   112             ; NUMBERS NOT EQUAL
022520                                  68$:
022520  000207                                  RTS     PC              ; RETURN TO TEST CALLER
                                        ;;*****************************************************************
                                        .SBTTL  SUBR TO TEST THE CLRD INSTRUCTION
022522                                  CLRDT:
022522  012700  000013                          MOV     #13,R0          ; LOAD $TMP0-12
022526  010501                                  MOV     R5,R1           ;   WITH TEST DATA SETS
022530  012702  001230                          MOV     #$TMP0,R2       ;   FOR DISPLAY LATER
022534  012122                                  MOV     (R1)+,(R2)+     ;
022536  077002                                  SOB     R0,.-2          ;
022540  012737  022546  001114                  MOV     #CLRDL,$LPERR   ; ERROR LOOPING ADDRESS
022546  170011                          CLRDL:  SETD                    ; D MODE
022550  011537  001170                          MOV     (R5),$REG0      ; INITIAL FLOAT NUMBER
022554  016537  000002  001172                  MOV     2(R5),$REG1     ;   INTO OPERATING ROOM
```

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

```
5564                                        .SBTTL  SUBR TO TEST THE TSTF INSTRUCTION
5565
5566  022732                                TSTFT:
5567  022732  012700  000007                        MOV     #7,R0           ; LOAD $TMP0-6
5568  022736  010501                              MOV     R5,R1           ;   WITH TEST DATA SETS
5569  022740  012702  001230                      MOV     #$TMP0,R2       ;   FOR DISPLAY LATER
5570  022744  012122                              MOV     (R1)+,(R2)+     ;
5571  022746  077002                              SOB     R0,.-2          ;
5572  022750  012737  022756  001114              MOV     #TSTFL,$LPERR   ; ERROR LOOPING ADDRESS
5573
5574  022756  170001                      TSTFL:  SETF                    ; F MODE
5575  022760  011537  001170                      MOV     (R5),$REG0      ; INITIAL FLOAT NUMBER
5576  022764  016537  000002  001172              MOV     2(R5),$REG1     ;   INTO OPERATING ROOM
5577  022772  170165  000010                      LDFPS   10(R5)          ; INITIAL FPS
5578
5579  022776  170537  001170              TSTFI:  TSTF    $REG0           ; ($REG0)->$REG0
5580
5581  023002  170237  002330                      STFPS   FPS             ; STORE FPS AFTER
5582  023006  170337  002332                      STST    FEC             ; STORE FEC/FEA AFTER
5583
5584  023012  023765  002330  000012              CMP     FPS,12(R5)      ; CHECK FPS
5585  023020  001401                              BEQ     65$             ; FPS IS OK
5586  023022  104076                              ERROR   76              ; FPS BAD
5587  023024  005765  000014              65$:    TST     14(R5)          ; DOES FEC/FEA APPLY?
5588  023030  100014                              BPL     66$             ; NO - SKIP TEST
5589  023032  012737  022776  002344              MOV     #TSTFI,EXPFEA   ; GET EXPECTED FEA
5590  023040  123765  002332  000014              CMPB    FEC,14(R5)      ; COMPARE FEC-S
5591  023046  001004                              BNE     64$             ; NOT EQUAL
5592  023050  023737  002334  002344              CMP     FEA,EXPFEA      ; COMPARE FEA-S
5593  023056  001401                              BEQ     66$             ; FEC, FEA OK
5594  023060  104101                      64$:    ERROR   101             ; FEC OR FEA ARE BAD
5595  023062                              66$:
5596
5597  023062  023765  001170  000004              CMP     $REG0,4(R5)     ; 1ST WORD OF RESULT CHECK?
5598  023070  001004                              BNE     67$             ; NO
5599  023072  023765  001172  000006              CMP     $REG1,6(R5)     ; 2ND WORD OF RESULT CHECK?
5600  023100  001401                              BEQ     68$             ; ALL WORDS OK
5601  023102  104114                      67$:    ERROR   114             ; NUMBERS NOT EQUAL
5602  023104                              68$:
5603
5604  023104  000207                              RTS     PC              ; RETURN TO TEST CALLER
5605
5606                                      ;;*****************************************************************
5607                                      .SBTTL  SUBR TO TEST THE TSTD INSTRUCTION
5608
5609  023106                              TSTDT:
5610  023106  012700  000013                      MOV     #13,R0          ; LOAD $TMP0-12
5611  023112  010501                              MOV     R5,R1           ;   WITH TEST DATA SETS
5612  023114  012702  001230                      MOV     #$TMP0,R2       ;   FOR DISPLAY LATER
5613  023120  012122                              MOV     (R1)+,(R2)+     ;
5614  023122  077002                              SOB     R0,.-2          ;
5615  023124  012737  023132  001114              MOV     #TSTDL,$LPERR   ; ERROR LOOPING ADDRESS
5616
5617  023132  170011                      TSTDL:  SETD                    ; D MODE
5618  023134  011537  001170                      MOV     (R5),$REG0      ; INITIAL FLOAT NUMBER
5619  023140  016537  000002  001172              MOV     2(R5),$REG1     ;   INTO OPERATING ROOM
```

```
5620  023146  016537  000004  001174          MOV     4(R5),$REG2     ;
5621  023154  016537  000006  001176          MOV     6(R5),$REG3     ;
5622  023162  170165  000020                  LDFPS   20(R5)          ; INITIAL FPS
5623
5624  023166  170537  001170          TSTDI:  TSTD    $REG0           ; ($REG0)->$REG0
5625
5626  023172  170237  002330                  STFPS   FPS             ; STORE FPS AFTER
5627  023176  170337  002332                  STST    FEC             ; STORE FEC/FEA AFTER
5628
5629  023202  023765  002330  000022          CMP     FPS,22(R5)      ; CHECK FPS
5630  023210  001401                          BEQ     65$             ; FPS IS OK
5631  023212  104077                          ERROR   77              ; FPS BAD
5632  023214  005765  000024          65$:    TST     24(R5)          ; DOES FEC/FEA APPLY?
5633  023220  100014                          BPL     66$             ; NO - SKIP TEST
5634  023222  012737  023166  002344          MOV     #TSTDI,EXPFEA   ; GET EXPECTED FEA
5635  023230  123765  002332  000024          CMPB    FEC,24(R5)      ; COMPARE FEC-S
5636  023236  001004                          BNE     64$             ; NOT EQUAL
5637  023240  023737  002334  002344          CMP     FEA,EXPFEA      ; COMPARE FEA-S
5638  023246  001401                          BEQ     66$             ; FEC, FEA OK
5639  023250  104102                  64$:    ERROR   102             ; FEC OR FEA ARE BAD
5640  023252                          66$:
5641
5642  023252  023765  001170  000010          CMP     $REG0,10(R5)    ; 1ST WORD OF RESULT CHECK?
5643  023260  001014                          BNE     67$             ; NO
5644  023262  023765  001172  000012          CMP     $REG1,12(R5)    ; 2ND WORD OF RESULT CHECK?
5645  023270  001010                          BNE     67$             ; NO
5646  023272  023765  001174  000014          CMP     $REG2,14(R5)    ; 3RD WORD OF RESULT CHECK?
5647  023300  001004                          BNE     67$             ; NO
5648  023302  023765  001176  000016          CMP     $REG3,16(R5)    ; 4TH WORD OF RESULT CHECK?
5649  023310  001401                          BEQ     68$             ; ALL WORDS OK
5650  023312  104115                  67$:    ERROR   115             ; NUMBERS NOT EQUAL
5651  023314                          68$:
5652
5653  023314  000207                          RTS     PC              ; RETURN TO TEST CALLER
5654
```

```
5655                                    .SBTTL  FPP TRAP CATCHER
5656
5657   023316  010637  002342   FPPILT: MOV     SP,FPPOSP       ; GET NEW SP
5658   023322  012637  002336           MOV     (SP)+,FPPOPC    ; POP OLD PC FOR DISPLAY
5659   023326  012637  002340           MOV     (SP)+,FPPOPS    ; POP OLD PS FOR DISPLAY
5660   023332  170237  002330           STFPS   FPS             ; GET FPS
5661   023336  170337  002332           STST    FEC             ; GET FEC/FEA
5662   023342  104116                   ERROR   116             ; SIGNAL UNEXPECTED FPP TRAP
5663   023344  013746  002340           MOV     FPPOPS,-(SP)    ; PUSH PSW
5664   023350  013746  002336           MOV     FPPOPC,-(SP)    ; PUSH PC
5665   023354  000002                   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($TSTNM) INTO THE DISPLAY REG.(DISPLAY<15:0>)
                                ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
                                ;*SW14=1         LOOP ON TEST
                                ;*SW11=1         INHIBIT ITERATIONS
                                ;*SW09=1         LOOP ON ERROR
                                ;*SW08=1         LOOP ON TEST IN "$LPTST"
                                ;*CALL
                                ;*      SCOPE           ;;SCOPE=IOT

023356                          $SCOPE:
023356                          64$:
023356  032777  040000  155562  1$:     BIT     #BIT14,@SWR     ;;LOOP ON PRESENT TEST?
023364  001114                          BNE     $OVER           ;;YES IF SW14=1
                                ;#####START OF CODE FOR THE XOR TESTER#####
023366  000416                  $XTSTR: BR      6$              ;;IF RUNNING ON THE "XOR" TESTER CHANGE
                                                                ;;THIS INSTRUCTION TO A "NOP" (NOP=240)
023370  013746  000004                  MOV     @#ERRVEC,-(SP)  ;;SAVE THE CONTENTS OF THE ERROR VECTOR
023374  012737  023414  000004          MOV     #5$,@#ERRVEC    ;;SET FOR TIMEOUT
023402  005737  177060                  TST     @#177060        ;;TIME OUT ON XOR?
023406  012637  000004                  MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
023412  000463                          BR      $SVLAD          ;;GO TO THE NEXT TEST
023414  022626                  5$:     CMP     (SP)+,(SP)+     ;;CLEAR THE STACK AFTER A TIME OUT
023416  012637  000004                  MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
023422  000423                          BR      7$              ;;LOOP ON THE PRESENT TEST
023424                          6$:;#####END OF CODE FOR THE XOR TESTER#####
023424  032777  000400  155514          BIT     #BIT08,@SWR     ;;LOOP ON SPEC. TEST?
023432  001404                          BEQ     2$              ;;BR IF NO
023434  023737  001112  001102          CMP     $LPTST,$TSTNM   ;;ON THE RIGHT TEST?
023442  001465                          BEQ     $OVER           ;;BR IF YES
023444  005737  001104          2$:     TST     $ERFLG          ;;HAS AN ERROR OCCURRED?
023450  001421                          BEQ     3$              ;;BR IF NO
023452  023737  001122  001104          CMP     $ERMAX,$ERFLG   ;;MAX. ERRORS FOR THIS TEST OCCURRED?
023460  101015                          BHI     3$              ;;BR IF NO
023462  032777  001000  155456          BIT     #BIT09,@SWR     ;;LOOP ON ERROR?
023470  001404                          BEQ     4$              ;;BR IF NO
023472  013737  001114  001110  7$:     MOV     $LPERR,$LPADR   ;;SET LOOP ADDRESS TO LAST SCOPE
023500  000446                          BR      $OVER
023502  005037  001104          4$:     CLR     $ERFLG          ;;ZERO THE ERROR FLAG
023506  005037  001310                  CLR     $TIMES          ;;CLEAR THE NUMBER OF ITERATIONS TO MAKE
023512  000415                          BR      1$              ;;ESCAPE TO THE NEXT TEST
023514  032777  004000  155424  3$:     BIT     #BIT11,@SWR     ;;INHIBIT ITERATIONS?
023522  001011                          BNE     1$              ;;BR IF YES
023524  005737  001332                  TST     $PASS           ;;IF FIRST PASS OF PROGRAM
023530  001406                          BEQ     1$              ;;      INHIBIT ITERATIONS
023532  005237  001106                  INC     $ICNT           ;;INCREMENT ITERATION COUNT
023536  023737  001310  001106          CMP     $TIMES,$ICNT    ;;CHECK THE NUMBER OF ITERATIONS MADE
023544  002024                          BGE     $OVER           ;;BR IF MORE ITERATION REQUIRED
023546  012737  000001  001106  1$:     MOV     #1,$ICNT        ;;REINITIALIZE THE ITERATION COUNTER
023554  013737  023632  001310          MOV     $MXCNT,$TIMES   ;;SET NUMBER OF ITERATIONS TO DO
023562  005237  001102          $SVLAD: INC     $TSTNM          ;;COUNT TEST NUMBERS
023566  013737  001102  001330          MOV     $TSTNM,$TESTN   ;;SET TEST NUMBER IN APT MAILBOX
023574  011637  001110                  MOV     (SP),$LPADR     ;;SAVE SCOPE LOOP ADDRESS
```

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

```
                                .SBTTL  ERROR HANDLER ROUTINE

                                ;;*****************************************************************
                                ;*THIS ROUTINE WIL. INCREMENT THE ERROR FLAG AND THE ERROR COUNT,
                                ;*SAVE THE ERROR ITEM NUMBER AND THE ADDRESS OF THE ERROR CALL
                                ;*AND GO TO $TYPERR 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

023634                          $ERROR:
023634  010037  002346                  MOV     R0,EREG0        ; DISPLAY R0
023640  010137  002350                  MOV     R1,EREG1        ;         R1
023644  010237  002352                  MOV     R2,EREG2        ;         R2
023650  010337  002354                  MOV     R3,EREG3        ;         R3
023654  010437  002356                  MOV     R4,EREG4        ;         R4
023660  010537  002360                  MOV     R5,EREG5        ;         R5
023664  010637  002362                  MOV     R6,EREG6        ; GET R6(SP) BEFORE TRAP
023670  062737  000004  002362          ADD     #4,EREG6        ;
023676  011637  002364                  MOV     (SP),EREG7      ; PC -> ERROR CALL INSTR
023702  005237  001104          7$:     INC     $ERFLG          ;;SET THE ERROR FLAG
023706  001775                  BEQ     7$              ;;DON'T LET THE FLAG GO TO ZERO
023710  013777  001102  155232  MOV     $TSTNM,@DISPLAY ;;DISPLAY TEST NUMBER
023716  032777  002000  155222  BIT     #BIT10,@SWR     ;;BELL ON ERROR?
023724  001402                  BEQ     1$              ;;NO - SKIP
023726  104401  001314          TYPE    ,$BELL          ;;RING BELL
023732  005237  001116          1$:     INC     $ERTTL          ;;COUNT THE NUMBER OF ERRORS
023736  011637  001124          MOV     (SP),$ERRPC     ;;GET ADDRESS OF ERROR INSTRUCTION
023742  162737  000002  001124  SUB     #2,$ERRPC
023750  117737  155150  001120  MOVB    @$ERRPC,$ITEMB  ;;STRIP AND SAVE THE ERROR ITEM CODE
023756  032777  020000  155162  BIT     #BIT13,@SWR     ;;SKIP TYPEOUT IF SET
023764  001004                  BNE     20$             ;;SKIP TYPEOUTS
023766  004737  024076          JSR     PC,$TYPERR      ;;GO TO USER ERROR ROUTINE
023772  104401  001321          TYPE    ,$CRLF
023776                          20$:
023776  122737  000001  001344  CMPB    #APTENV,$ENV    ;;RUNNING IN APT MODE
024004  001007                  BNE     2$              ;;NO,SKIP APT ERROR REPORT
024006  113737  001120  024020  MOVB    $ITEMB,21$      ;;SET ITEM NUMBER AS ERROR NUMBER
024014  004737  024612          JSR     PC,$ATY4        ;;REPORT FATAL ERROR TO APT
024020      000                 21$:    .BYTE   0
024021      000                 .BYTE   0
024022  000777                  22$:    BR      22$             ;;APT ERROR LOOP
024024  005777  155116          2$:     TST     @SWR            ;;HALT ON ERROR
024030  100001                  BPL     3$              ;;SKIP IF CONTINUE
024032  000000                  HALT                    ;;HALT ON ERROR!
024034  032777  001000  155104  3$:     BIT     #BIT09,@SWR     ;;LOOP ON ERROR SWITCH SET?
024042  001402                  BEQ     4$              ;;BR IF NO
024044  013716  001114          MOV     $LPERR,(SP)     ;;FUDGE RETURN FOR LOOPING
024050  005737  001312          4$:     TST     $ESCAPE         ;;CHECK FOR AN ESCAPE ADDRESS
024054  001402                  BEQ     5$              ;;BR IF NONE
024056  013716  001312          MOV     $ESCAPE,(SP)    ;;FUDGE RETURN ADDRESS FOR ESCAPE
024062                          5$:
```

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

```
                        ;;*****************************************************************
                        .SBTTL  ERROR MESSAGE TYPEOUT ROUTINE (MODIFIED SYSMAC)

                        ;*THIS RCUTINE 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 $ERRTYP, 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).

024076                  $TYPERR:
024076  104401                  TYPE                    ; TYPE "HOT" OR "WARM"
024100  001321          HOTWRM: .WORD   $CRLF           ; PTR TO MESSAGE
024102  010046                  MOV     R0,-(SP)        ; SAVE R0
024104  010146                  MOV     R1,-(SP)        ; SAVE R1
024106  005000                  CLR     R0              ; PICKUP ITEM INDEX
024110  153700  001120          BISB    @#$ITEMB,R0
024114  001004                  BNE     1$              ; IF ITEM NUMBER FROM ERROR 0,
                                                        ;   JUST TYPE PC OF ERROR
024116  013746  001124          MOV     $ERRPC,-(SP)    ; GET ERROR PC FOR TYPEOUT
024122  104402                  TYPOC                   ; TYPE OCTAL, ALL DIGITS
024124  000452                  BR      7$              ; EXIT
024126  005300          1$:     DEC     R0              ; ADJUST ERROR # FOR TABLE INDEX
024130  006300                  ASL     R0              ;   OF 6 BYTES/ENTRY
024132  010001                  MOV     R0,R1
024134  006300                  ASL     R0
024136  060100                  ADD     R1,R0
024140  062700  001354          ADD     #$ERRTB,R0      ; FORM TABLE PTR
024144  012037  024154          MOV     (R0)+,2$        ; PICKUP "ERROR MESSAGE" PTR
024150  001404                  BEQ     3$              ; SKIP TYPEOUT IF NULL
024152  104401                  TYPE                    ; TYPE "ERROR MESSAGE"
024154  000000          2$:     .WORD   0               ; "ERROR MESSAGE" PTR HERE
024156  104401  001321          TYPE    ,$CRLF          ; CR & LF
024162  104401  024272  3$:     TYPE    ,11$            ; "TEST #    ERR PC" HEADER
024166  012037  024176          MOV     @(R0)+,4$       ; PICKUP "DATA HEADER" PTR
024172  001402                  BEQ     5$              ; SKIP TYPEOUT IF NULL
024174  104401                  TYPE                    ; TYPE "DATA HEADER"
024176  000000          4$:     .WORD   0               ; "DATA HEADER" PTR HERE
024200  104401  001321  5$:     TYPE    ,$CRLF          ; CR & LF
024204  017746  000054          MOV     @$TESTN,-(SP)   ; ($TESTN)
024210  104402                  TYPOC                   ; OCTAL W/ LEADING ZEROS
024212  104401  024270          TYPE    ,10$            ; <HT>
024216  017746  000044          MOV     @$ERRPC,-(SP)   ; ($ERRPC)
024222  104402                  TYPOC                   ; OCTAL W/ LEADING ZEROS
024224  104401  024270          TYPE    ,10$            ; <HT>
024230  011000                  MOV     @(R0),R0        ; PICKUP "DATA TABLE" PTR
024232  001407                  BEQ     7$              ; EXIT IF NULL
024234  013046          6$:     MOV     @(R0)+,-(SP)    ; SAVE ... FOR TYPEOUT
024236  104402                  TYPOC                   ; TYPE OCTAL, ALL DIGITS
024240  005710                  TST     (R0)            ; ANOTHER NUMBER ?
024242  001403                  BEQ     7$              ; NO - EXIT
024244  104401  024270          TYPE    ,10$            ; TAB BETWEEN ELEMENTS
```

```
5846  024250  000771                        BR      6$              ; LOOP ON DATA TABLE VECTOR
5847  024252  012601                7$:     MOV     (SP)+,R1        ; RESTORE R1
5848  024254  012600                        MOV     (SP)+,R0        ; RESTORE R0
5849  024256  104401  001321              TYPE    ,$CRLF          ; CR & LF
5850  024262  000207                        RTS     PC              ; RETURN
5851  024264  001330                8$:     .WORD   $TESTN          ;
5852  024266  001124                9$:     .WORD   $ERRPC          ;
5853  024270  000011                10$:    .ASCIZ  <11>            ; <HT>
5854  024272  042524  052123  021440 11$:    .ASCIZ  "TEST # ERR PC "
5855  024300  042411  051122  050040
5856  024306  004503     000
5857          024312                        .EVEN
```

```
                                .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
                                ;*

024312  105737  001165          $TYPE:  TSTB    $TPFLG          ;;IS THERE A TERMINAL?
024316  100002                          BPL     1$              ;;BR IF YES
024320  000000                          HALT                    ;;HALT HERE IF NO TERMINAL
024322  000430                          BR      3$              ;;LEAVE
024324  010046                  1$:     MOV     R0,-(SP)        ;;SAVE R0
024326  017600  000002                  MOV     @2(SP),R0       ;;GET ADDRESS OF ASCIZ STRING
024332  122737  000001  001344          CMPB    #APTENV,$ENV    ;;RUNNING IN APT MODE
024340  001011                          BNE     62$             ;;NO,GO CHECK FOR APT CONSOLE
024342  132737  000100  001345          BITB    #APTSPOOL,$ENVM ;;SPOOL MESSAGE TO APT
024350  001405                          BEQ     62$             ;;NO,GO CHECK FOR CONSOLE
024352  010037  024362                  MOV     R0,61$          ;;SETUP MESSAGE ADDRESS FOR APT
024356  004737  024602                  JSR     PC,$ATY3        ;;SPOOL MESSAGE TO APT
024362  000000                  61$:    .WORD   0               ;;MESSAGE ADDRESS
024364  132737  000040  001345  62$:    BITB    #APTCSUP,$ENVM  ;;APT CONSOLE SUPPRESSED
024372  001003                          BNE     60$             ;;YES,SKIP TYPE OUT
024374  112046                  2$:     MOVB    (R0)+,-(SP)     ;;PUSH CHARACTER TO BE TYPED ONTO STACK
024376  001005                          BNE     4$              ;;BR IF IT ISN'T THE TERMINATOR
024400  005726                          TST     (SP)+           ;;IF TERMINATOR POP IT OFF THE STACK
024402  012600                  60$:    MOV     (SP)+,R0        ;;RESTORE R0
024404  062716  000002          3$:     ADD     #2,(SP)         ;;ADJUST RETURN PC
024410  000002                          RTI                     ;;RETURN
024412  122716  000011          4$:     CMPB    #HT,(SP)        ;;BRANCH IF <HT>
024416  001430                          BEQ     8$
024420  122716  000200                  CMPB    #CRLF,(SP)      ;;BRANCH IF NOT <CRLF>
024424  001006                          BNE     5$
024426  005726                          TST     (SP)+           ;;POP  <CR><LF> EQUIV
024430  104401                          TYPE                    ;;TYPE A CR AND LF
024432  001321                          $CRLF
024434  105037  024570                  CLRB    $CHARCNT        ;;CLEAR CHARACTER COUNT
024440  000755                          BR      2$              ;;GET NEXT CHARACTER
024442  004737  024524          5$:     JSR     PC,$TYPEC       ;;GO TYPE THIS CHARACTER
024446  123726  001164          6$:     CMPB    $FILLC,(SP)+    ;;IS IT TIME FOR FILLER CHARS.?
024452  001350                          BNE     2$              ;;IF NO GO GET NEXT CHAR.
024454  013746  001162                  MOV     $NULL,-(SP)     ;;GET # OF FILLER CHARS. NEEDED
                                                                ;;AND THE NULL CHAR.
024460  105366  000001          7$:     DECB    1(SP)           ;;DOES A NULL NEED TO BE TYPED?
024464  002770                          BLT     6$              ;;BR IF NO--GO POP THE NULL OFF OF STACK
024466  004737  024524                  JSR     PC,$TYPEC       ;;GO TYPE A NULL
024472  105337  024570                  DECB    $CHARCNT        ;;DO NOT COUNT AS A COUNT
```

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

```
                                        .SBTTL  APT COMMUNICATIONS ROUTINE
                                        ;;*****************************************************************
024574  112737  000001  025040  $ATY1:  MOVB    #1,$FFLG        ;;TO REPORT FATAL ERROR
024602  112737  000001  025036  $ATY3:  MOVB    #1,$MFLG        ;;TO TYPE A MESSAGE
024610  000403                  $ATY4:  BR      $ATYC
024612  112737  000001  025040  $ATY4:  MOVB    #1,$FFLG        ;;TO ONLY REPORT FATAL ERROR
024620                          $ATYC:
024620  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
024622  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
024624  105737  025036                  TSTB    $MFLG           ;;SHOULD TYPE A MESSAGE?
024630  001450                          BEQ     5$              ;;IF NOT:  BR
024632  122737  000001  001344          CMPB    #APTENV,$ENV    ;;OPERATING UNDER APT?
024640  001031                          BNE     3$              ;;IF NOT:  BR
024642  132737  000100  001345          BITB    #APTSPOOL,$ENVM ;;SHOULD SPOOL MESSAGES?
024650  001425                          BEQ     3$              ;;IF NOT:  BR
024652  017600  000004                  MOV     @4(SP),R0       ;;GET MESSAGE ADDR.
024656  062766  000002  000004          ADD     #2,4(SP)                ;;BUMP RETURN ADDR.
024664  005737  001324          1$:     TST     $MSGTYPE        ;;SEE IF DONE W/ LAST XMISSION?
024670  001375                          BNE     1$              ;;IF NOT:  WAIT
024672  010037  001340                  MOV     R0,$MSGAD       ;;PUT ADDR IN MAILBOX
024676  105720                  2$:     TSTB    (R0)+           ;;FIND END OF MESSAGE
024700  001376                          BNE     2$
024702  163700  001340                  SUB     $MSGAD,R0       ;;SUB START OF MESSAGE
024706  006200                          ASR     R0              ;;GET MESSAGE LNGTH IN WORDS
024710  010037  001342                  MOV     R0,$MSGLGT      ;;PUT LENGTH IN MAILBOX
024714  012737  000004  001324          MOV     #4,$MSGTYPE     ;;TELL APT TO TAKE MSG.
024722  000413                          BR      5$
024724  017637  000004  024750  3$:     MOV     @4(SP),4$       ;;PUT MSG ADDR IN JSR LINKAGE
024732  062766  000002  000004          ADD     #2,4(SP)                ;;BUMP RETURN ADDRESS
024740  013746  177776                  MOV     177776,-(SP)    ;;PUSH 177776 ON STACK
024744  004737  024312                  JSR     PC,$TYPE        ;;CALL TYPE MACRO
024750  000000                  4$:     .WORD   0
024752                          5$:
024752  105737  025040          10$:    TSTB    $FFLG           ;;SHOULD REPORT FATAL ERROR?
024756  001416                          BEQ     12$             ;;IF NOT:  BR
024760  005737  001344                  TST     $ENV            ;;RUNNING UNDER APT?
024764  001413                          BEQ     12$             ;;IF NOT:  BR
024766  005737  001324          11$:    TST     $MSGTYPE        ;;FINISHED LAST MESSAGE?
024772  001375                          BNE     11$             ;;IF NOT:  WAIT
024774  017637  000004  001326          MOV     @4(SP),$FATAL   ;;GET ERROR #
025002  062766  000002  000004          ADD     #2,4(SP)                ;;BUMP RETURN ADDR.
025010  005237  001324                  INC     $MSGTYPE        ;;TELL APT TO TAKE ERROR
025014  105037  025040          12$:    CLRB    $FFLG           ;;CLEAR FATAL FLAG
025020  105037  025037                  CLRB    $LFLG           ;;CLEAR LOG FLAG
025024  105037  025036                  CLRB    $MFLG           ;;CLEAR MESSAGE FLAG
025030  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
025032  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
025034  000207                          RTS     PC              ;;RETURN
025036     000          $MFLG:  .BYTE   0               ;;MESSG. FLAG
025037     000          $LFLG:  .BYTE   0               ;;LOG FLAG
025040     000          $FFLG:  .BYTE   0               ;;FATAL FLAG
025042                          .EVEN
000200                  APTSIZE=200
000001                  APTENV=001
000100                  APTSPOOL=100
```

```
  5993          000040                  APTCSUP=040
```

```
                                .SBTTL  BINARY TO OCTAL (ASCII) AND TYPE

                                ;;**********************************************************************
                                ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIGIT
                                ;*OCTAL (ASCII) NUMBER AND TYPE IT.
                                ;*$TYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
                                ;*CALL:
                                ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
                                ;*      TYPOS                   ;;CALL FOR TYPEOUT
                                ;*      .BYTE   N               ;;N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
                                ;*      .BYTE   M               ;;M=1 OR 0
                                ;*                                      ;;1=TYPE LEADING ZEROS
                                ;*                                      ;;0=SUPPRESS LEADING ZEROS
                                ;*
                                ;*$TYPON----ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST
                                ;*$TYPOS OR $TYPOC
                                ;*CALL:
                                ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
                                ;*      TYPON                   ;;CALL FOR TYPEOUT
                                ;*
                                ;*$TYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
                                ;*CALL:
                                ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
                                ;*      TYPOC                   ;;CALL FOR TYPEOUT

025042  017646  000000          $TYPOS: MOV     @(SP),-(SP)     ;;PICKUP THE MODE
025046  116637  000001  025265          MOVB    1(SP),$OFILL    ;;LOAD ZERO FILL SWITCH
025054  112637  025267                  MOVB    (SP)+,$OMODE+1  ;;NUMBER OF DIGITS TO TYPE
025060  062716  000002                  ADD     #2,(SP)         ;;ADJUST RETURN ADDRESS
025064  000406                          BR      $TYPON
025066  112737  000001  025265  $TYPOC: MOVB    #1,$OFILL       ;;SET THE ZERO FILL SWITCH
025074  112737  000006  025267          MOVB    #6,$OMODE+1     ;;SET FOR SIX(6) DIGITS
025102  112737  000005  025264  $TYPON: MOVB    #5,$OCNT        ;;SET THE ITERATION COUNT
025110  010346                          MOV     R3,-(SP)        ;;SAVE R3
025112  010446                          MOV     R4,-(SP)        ;;SAVE R4
025114  010546                          MOV     R5,-(SP)        ;;SAVE R5
025116  113704  025267                  MOVB    $OMODE+1,R4     ;;GET THE NUMBER OF DIGITS TO TYPE
025122  005404                          NEG     R4
025124  062704  000006                  ADD     #6,R4           ;;SUBTRACT IT FOR MAX. ALLOWED
025130  110437  025266                  MOVB    R4,$OMODE       ;;SAVE IT FOR USE
025134  113704  025265                  MOVB    $OFILL,R4       ;;GET THE ZERO FILL SWITCH
025140  016605  000012                  MOV     12(SP),R5       ;;PICKUP THE INPUT NUMBER
025144  005003                          CLR     R3              ;;CLEAR THE OUTPUT WORD
025146  006105                  1$:     ROL     R5              ;;ROTATE MSB INTO "C"
025150  000404                          BR      3$              ;;GO DO MSB
025152  006105                  2$:     ROL     R5              ;;FORM THIS DIGIT
025154  006105                          ROL     R5
025156  006105                          ROL     R5
025160  010503                          MOV     R5,R3
025162  006103                  3$:     ROL     R3              ;;GET LSB OF THIS DIGIT
025164  105337  025266                  DECB    $OMODE          ;;TYPE THIS DIGIT?
025170  100016                          BPL     7$              ;;BR IF NO
025172  042703  177770                  BIC     #177770,R3      ;;GET RID OF JUNK
025176  001002                          BNE     4$              ;;TEST FOR 0
025200  005704                          TST     R4              ;;SUPPRESS THIS 0?
025202  001403                          BEQ     5$              ;;BR IF YES
```

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

```
                                        .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.

025270  010046                  $TRAP:  MOV     R0,-(SP)        ;;SAVE R0
025272  016600  000002                  MOV     2(SP),R0        ;;GET TRAP ADDRESS
025276  005740                          TST     -(R0)           ;;BACKUP BY 2
025300  111000                          MOVB    (R0),R0         ;;GET RIGHT BYTE OF TRAP
025302  006300                          ASL     R0              ;;POSITION FOR INDEXING
025304  016000  025324                  MOV     $TRPAD(R0),R0   ;;INDEX TO TABLE
025310  000200                          RTS     R0              ;;GO TO ROUTINE

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

025312  011646                  $TRAP2: MOV     (SP),-(SP)      ;;MOVE THE PC DOWN
025314  016666  000004  000002          MOV     4(SP),2(SP)     ;;MOVE THE PSW DOWN
025322  000002                          RTI                     ;;RESTORE THE PSW

                                        .SBTTL  TRAP TABLE

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

                                        ;       ROUTINE
                                        ;       -------
025324  025312                  $TRPAD: .WORD   $TRAP2
025326  024312                          $TYPE   ;;CALL=TYPE     TRAP+1(104401)  TTY TYPEOUT ROUTINE
025330  025066                          $TYPOC  ;;CALL=TYPOC    TRAP+2(104402)  TYPE OCTAL NUMBER (WITH LEADING ZEROS)
025332  025042                          $TYPOS  ;;CALL=TYPOS    TRAP+3(104403)  TYPE OCTAL NUMBER (NO LEADING ZEROS)
025334  025102                          $TYPON  ;;CALL=TYPON    TRAP+4(104404)  TYPE OCTAL NUMBER (AS PER LAST CALL)
```

```
                                        .SBTTL  POWER DOWN AND UP ROUTINES

                                        ;;*****************************************************************
                                        ;POWER DOWN ROUTINE
025336  012737  025510  000024  $PWRON: MOV     #$ILLUP,@#PWRVEC ;;SET FOR FAST UP
025344  012737  000340  000026          MOV     #340,@#PWRVEC+2 ;;PRIO:7
025352  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
025354  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
025356  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
025360  010346                          MOV     R3,-(SP)        ;;PUSH R3 ON STACK
025362  010446                          MOV     R4,-(SP)        ;;PUSH R4 ON STACK
025364  010546                          MOV     R5,-(SP)        ;;PUSH R5 ON STACK
025366  017746  153554                  MOV     @$WR,-(SP)      ;;PUSH @$WR ON STACK
025372  010637  025514                  MOV     SP,$SAVR6       ;;SAVE SP
025376  012737  025410  000024          MOV     #$PWRUP,@#PWRVEC ;;SET UP VECTOR
025404  000000                          HALT
025406  000776                          BR      .-2             ;;HANG UP

                                        ;;*****************************************************************
                                        ;POWER UP ROUTINE
025410  012737  025510  000024  $PWRUP: MOV     #$ILLUP,@#PWRVEC ;;SET FOR FAST DOWN
025416  013706  025514                  MOV     $SAVR6,SP       ;;GET SP
025422  005037  025514                  CLR     $SAVR6          ;;WAIT LOOP FOR THE TTY
025426  005237  025514          1$:     INC     $SAVR6          ;;WAIT FOR THE INC
025432  001375                          BNE     1$              ;;OF  WORD
025434  011600                          MOV     (SP),R0         ;GET SAVED $WR OFF STACK
025436  076600  000226                  MED     ,226            ;RESTORE $WR CONTENTS
025442  012677  153500                  MOV     (SP)+,@$WR      ;;POP STACK INTO @$WR
025446  012605                          MOV     (SP)+,R5        ;;POP STACK INTO R5
025450  012604                          MOV     (SP)+,R4        ;;POP STACK INTO R4
025452  012603                          MOV     (SP)+,R3        ;;POP STACK INTO R3
025454  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
025456  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
025460  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
025462  012737  025336  000024          MOV     #$PWRON,@#PWRVEC ;;SET UP THE POWER DOWN VECTOR
025470  012737  000340  000026          MOV     #340,@#PWRVEC+2 ;;PRIO:7
025476  104401                          TYPE                    ;REPORT THE POWER FAILURE
025500  025516                  $PWRMG: .WORD   $POWER          ;;POWER FAIL MESSAGE POINTER
025502  012716                          MOV     (PC)+,(??)      ;;RESTART AT START
025504  002506                  $PWRAD: .WORD   START           ;;RESTART ADDRESS
025506  000002                          RTI
025510  000000                  $ILLUP: HALT                    ;;THE POWER UP SEQUENCE WAS STARTED
025512  000776                          BR      .-2             ;; BEFORE THE POWER DOWN WAS COMPLETE
025514  000000                  $SAVR6: 0                       ;;PUT THE SP HERE
025516  005015  047520  042527  $POWER: .ASCIZ  <15><12>"POWER"
025524  000122
                                        .EVEN
```

```
6155                                            .SBTTL  ERROR MESSAGES, DATA HEADERS, DATA VECTORS, ETC
6156
6157                                                    ; MESSAGE PREFIXES
6158  025526  047510  035124  000040  ASCHOT: .ASCIZ  "HOT: "
6159  025534  040527  046522  020072  ASCWRM: .ASCIZ  "WARM: "
6160  025542     000
6161
6162                                                    ; ERROR MESSAGES HERE
6163  025543     122  041505  044505  EMA:    .ASCIZ  "RECEIVED FPS IS BAD"
6164  025550  042526  020104  053106
6165  025556  020123  051511  041040
6166  025564  042101     000
6167  025567     122  041505  044505  EMB:    .ASCIZ  "RECEIVED FEC/FEA IS BAD"
6168  025574  042526  020104  042506
6169  025602  027503  042506  020101
6170  025610  051511  041040  042101
6171  025616     000
6172  025617     101  051502  043050  EMC:    .ASCIZ  "ABS(F/D) OPERATION - RESULT INCORRECT"
6173  025624  042057  020051  050117
6174  025632  051105  052101  047511
6175  025640  020116  020055  042522
6176  025646  052523  052114  044440
6177  025654  041516  051117  042522
6178  025662  052103     000
6179  025665     125  042516  050130  EMD:    .ASCIZ  "UNEXPECTED FLOATING POINT TRAP, IGNORED & CONTINUING"
6180  025672  041505  042524  020104
6181  025700  046106  040517  044524
6182  025706  043516  050040  044517
6183  025714  052116  052040  040522
6184  025722  026120  044440  047107
6185  025730  051117  042105  023040
6186  025736  041''40  047117  044524
6187  025744  052516  047111  000107
6188  025752  047125  041101  042514  EME:    .ASCIZ  "UNABLE TO REFERENCE FLOATING ACCUMULATOR"
6189  025760  052040  02''17  042522
6190  025766  042506  042522  041516
6191  025774  020105  046106  040517
6192  026002  044524  043516  040440
6193  026010  041503  046525  046125
6194  026016  052101  051117     000
6195  026023     122  050111  046120  EMF:    .ASCIZ  "RIPPLING A 1 THRU FPS FAILED"
6196  026030  047111  020107  020101
6197  026036  020061  044124  052522
6198  026044  043040  051520  043040
6199  026052  044501  042514  000104
6200  026060  044522  050120  044514  EMG:    .ASCIZ  "RIPPLING A 0 THRU FPS FAILED"
6201  026066  043516  040440  030040
6202  026074  052040  051110  020125
6203  026102  050106  020123  040506
6204  026110  046111  042105     000
6205  026115     123  052105  020106  EMH:    .ASCIZ  "SETF FAILED TO CLEAR FPS BIT 7"
6206  026122  040506  046111  042105
6207  026130  052040  020117  046103
6208  026136  040505  020122  050106
6209  026144  020123  044502  020124
6210  026152  000067
```

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

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

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

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

```
                                                        ; DATA HEADERS HERE
030215    114  040517  042504  DHA:    .ASCIZ  "LOADED STORED"
030222 004504  052123  051117
030230 042105     000
030233    123  047524  042522  DHB:    .ASCIZ  "STORED"
030240 000104
030242 026455  026455  047514  DHC:    .ASCIZ  "----LOADED---- ----STORED----"
030250 042101  042105  026455
030256 026455  026411  026455
030264 051455  047524  042522
030272 026504  026455  000055
030300 054105  023520  004504  DHD:    .ASCIZ  "EXP'D  RCV'D"
030306 041522  023526  000104
030314 054105  023520  026504  DHE:    .ASCIZ  "EXP'D-FEC-RCV'D        EXP'D-FEA-RCV'D"
030322 042506  026503  041522
030330 023526  004504  054105
030336 023520  026504  042506
030344 026501  041522  023526
030352 000104
030354 026455  042455  050130  DHF:    .ASCIZ  "---EXPECTED--- ---RECEIVED---"
030362 041505  047524  026504
030370 026455  026411  026455
030376 042522  042503  053111
030404 042105  026455  000055
030412 026455  026455  026455  DHG:    .ASCII  "----------EXPECTED----------- "
030420 026455  026455  042455
030426 050130  041505  042524
030434 026504  026455  026455
030442 026455  026455  026455
030450     011
030451     055  026455  026455          .ASCIZ  "----------RECEIVED----------"
030456 026455  026455  026455
030464 042522  042503  053111
030472 042105  026455  026455
030500 026455  026455  026455
030506 000055
030510 046117  020104  041520  DHH:    .ASCIZ  "OLD PC OLD PS  NEW SP   FPS     FEC     FEA"
030516 047411  042114  050040
030524 004523  042516  020127
030532 050123  020011  050106
030540 004523  043040  041505
030546 020011  042506  000101
030554 047514  042101  042105  DHI:    .ASCIZ  "LOADED STORED    R2"
030562 051411  047524  042522
030570 004504  020040  031122
030576     000
030577    114  040517  042504  DHJ:    .ASCIZ  "LOADED STORED    R3"
030604 004504  052123  051117
030612 042105  020011  051040
030620 000063
030622 047514  042101  042105  DHK:    .ASCIZ  "LOADED STORED    R0      R2"
030630 051411  047524  042522
030636 004504  020040  030122
030644 020011  051040  000062
030652 047514  042101  042105  DHL:    .ASCIZ  "LOADED STORED  $REG2     R0"
030660 051411  047524  042522
```

```
6462  030666  004504  051044  043505
6463  030674  004462  020040  030122
6464  030702     000
6465  030703     114  040517  042504  DHM:    .ASCIZ  "LOADED STORED  $REG2     R2"
6466  030710  004504  052123  051117
6467  030716  042105  022011  042522
6468  030724  031107  020011  051040
6469  030732  000062
6470  030734  047514  042101  042105  DHN:    .ASCIZ  "LOADED STORED  $REG2     R4"
6471  030742  051411  047524  042522
6472  030750  004504  051044  043505
6473  030756  004462  020040  032122
6474  030764     000
6475  030765     114  040517  042504  DHO:    .ASCIZ  "LOADED STORED  $REG2     R5"
6476  030772  004504  052123  051117
6477  031000  042105  022011  042522
6478  031006  031107  020011  051040
6479  031014  000065
6480  031016  047514  042101  042105  DHP:    .ASCIZ  "LOADED STORED  $REG2     R1      R3"
6481  031024  051411  047524  042522
6482  031032  004504  051044  043505
6483  031040  004462  020040  030522
6484  031046  020011  051040  000063
6485  031054  047514  042101  042105  DHQ:    .ASCIZ  "LOADED STORED  $REG2     R3      R5"
6486  031062  051411  047524  042522
6487  031070  004504  051044  043505
6488  031076  004462  020040  031522
6489  031104  020011  051040  000065
6490  031112  026455  026455  026455  DHR:    .ASCIZ  "------------STORED------------"
6491  031120  026455  026455  026455
6492  031126  052123  051117  042105
6493  031134  026455  026455  026455
6494  031142  026455  026455  026455
6495  031150     000
6496  031151     055  026455  026455  DHS:    .ASCIZ  "------------STORED------------    R4"
6497  031156  026455  026455  026455
6498  031164  051455  047524  042522
6499  031172  026504  026455  026455
6500  031200  026455  026455  026455
6501  031206  004455  020040  032122
6502  031214     000
6503  031215     055  026455  026455  DHT:    .ASCIZ  "------------STORED------------    R5"
6504  031222  026455  026455  026455
6505  031230  051455  047524  042522
6506  031236  026504  026455  026455
6507  031244  026455  026455  026455
6508  031252  004455  020040  032522
6509  031260     000
6510  031261     055  026455  026455  DHU:    .ASCII  "------------STORED------------    R0"
6511  031266  026455  026455  026455
6512  031274  051455  047524  042522
6513  031302  026504  026455  026455
6514  031310  026455  026455  026455
6515  031316  004455  020040  030122
6516  031324  020011  051040  000063          .ASCIZ  "          R3"
6517  031332  026455  026455  026455  DHV:    .ASCII  "------------STORED------------    R1"
```

```
6518  031340  026455  026455  026455
6519  031346  052123  051117  042105
6520  031354  026455  026455  026455
6521  031362  026455  026455  026455
6522  031370  020011  051040      061
6523  031375      040  004440  031522          .ASCIZ  "       R3"
6524  031402      000
6525  031403      055  026455  026455  DHW:    .ASCII  "------------STORED------------ $REG10"
6526  031410  026455  026455  026455
6527  031416  051455  047524  042522
6528  031424  026504  026455  026455
6529  031432  026455  026455  026455
6530  031440  004455  051044  043505
6531  031446  030061
6532  031450  020040  051011  000060          .ASCIZ  "       R0"
6533  031456  026455  026455  026455  DHX:    .ASCII  "------------STORED------------ $REG10"
6534  031464  026455  026455  026455
6535  031472  052123  051117  042105
6536  031500  026455  026455  026455
6537  031506  026455  026455  026455
6538  031514  022011  042522  030507
6539  031522      060
6540  031523      040  004440  031122          .ASCIZ  "       R2"
6541  031530      000
6542  031531      055  026455  026455  DHY:    .ASCII  "------------STORED------------ $REG10"
6543  031536  026455  026455  026455
5544  031544  051455  047524  042522
6545  031552  026504  026455  026455
6546  031560  026455  026455  026455
6547  031566  004455  051044  043505
6548  031574  030061
6549  031576  020040  051011  000064          .ASCIZ  "       R4"
6550  031604  026455  026455  026455  DHZ:    .ASCII  "------------STORED------------ $REG10"
6551  031612  026455  026455  026455
6552  031620  052123  051117  042105
6553  031626  026455  026455  026455
6554  031634  026455  026455  026455
6555  031642  022011  042522  030507
6556  031650      060
6557  031651      011  051044  043505          .ASCIZ  "       $REG11    R1"
6558  031656  030461  020011  051040
6559  031664  000061
6560  031666  026455  026455  026455  DHAA:   .ASCII  "------------STORED------------ $REG10"
6561  031674  026455  026455  026455
6562  031702  052123  051117  042105
6563  031710  026455  026455  026455
6564  031716  026455  026455  026455
6565  031724  022011  042522  030507
6566  031732      060
6567  031733      011  051044  043505          .ASCIZ  "       $REG11    R4      R5"
6568  031740  030461  020011  051040
6569  031746  004464  020040  032522
6570  031754      000
6571  031755      055  026455  046055  DHAB:   .ASCIZ  "----LOADED---- ----STORED----    R1"
6572  031762  040517  042504  026504
6573  031770  026455  004455  026455
```

```
6574  031776  026455  052123  051117
6575  032004  042105  026455  026455
6576  032012  020011  051040  000061
6577  032020  026455  026455  047514  DHAC:   .ASCIZ  "----LOADED---- ----STORED----    R3"
6578  03    6  042101  042105  026455
6579  03    4  026455  026411  026455
6580  032042  051455  047524  042522
6581  03    0  026504  026455  004455
6582  03    6  020040  031522     000
6583  03    3     055  026455  046055  DHAD:   .ASCII  "----LOADED---- ----STORED----    R2"
6584  032070  040517  042504  026504
6585  032076  026455  004455  026455
6586  032104  026455  052123  051117
6587  032112  042105  026455  026455
6588  032120  020011  051040     062
6589  032125     011  020040  032522          .ASCIZ  "         R5"
6590  032132     000
6591  032133     055  026455  046055  DHAE:   .ASCIZ  "----LOADED---- ----STORED----  $REG4"
6592  032140  040517  042504  026504
6593  032146  026455  004455  026455
6594  032154  026455  052123  051117
6595  032162  042105  026455  026455
6596  032170  022011  042522  032107
6597  032176     000
6598  032177     055  026455  046055  DHAF:   .ASCII  "----LOADED---- ----STORED----  $REG4"
6599  032204  040517  042504  026504
6600  032212  026455  004455  026455
6601  032220  026455  052123  051117
6602  032226  042105  026455  026455
6603  032234  022011  042522  032107
6604  032242  020011  051040  004460          .ASCIZ  "         R0      R4"
6605  032250  020040  032122     000
6606  032255     055  026455  046055  DHAG:   .ASCII  "----LOADED---- ----STORED----  $REG4"
6607  032262  040517  042504  026504
6608  032270  026455  004455  026455
6609  032276  026455  052123  051117
6610  032304  042105  026455  026455
6611  032312  022011  042522  032107
6612  032320  020011  051040  004461          .ASCIZ  "         R1      R2"
6613  032326  020040  031122     000
6614  032333     055  026455  046055  DHAH:   .ASCII  "----LOADED---- ----STORED----  $REG4"
6615  032340  040517  042504  026504
6616  032346  026455  004455  026455
6617  032354  026455  052123  051117
6618  032362  042105  026455  026455
6619  032370  022011  042522  032107
6620  032376  020011  051040  004464          .ASCIZ  "         R4      R5"
6621  032404  020040  032522     000
6622  032411     055  026455  046055  DHAI:   .ASCII  "----LOADED---- ----STORED----  $REG4"
6623  032416  040517  042504  026504
6624  032424  026455  004455  026455
6625  032432  026455  052123  051117
6626  032440  042105  026455  026455
6627  032446  022011  042522  032107
6628  032454  022011  042522  032507          .ASCIZ  "       $REG5     R5"
6629  032462  020011  051040  000065
```

```
 6630  032470  026455  026455  047514  DHAJ:   .ASCII  "----LOADED---- ----STORED----  $REG4"
 6631  032476  042101  042105  026455
 6632  032504  026455  026411  026455
 6633  032512  051455  047524  042522
 6634  032520  026504  026455  004455
 6635  032526  051044  043505     064
 6636  032533     011  051044  043505          .ASCIZ  "       $REG5     R0      R2"
 6637  032540  004465  020040  030122
 6638  032546  020011  051040  000062
 6639  032554  026455  026455  047514  DHAK:   .ASCIZ  "----LOADED---- ----STORED----"
 6640  032562  042101  042105  026455
 6641  032570  026455  026411  026455
 6642  032576  051455  047524  042522
 6643  032604  026504  026455  000055
 6644  032612  000040                  DHAL:   .ASCIZ  " "
 6645
 6646
```

```
6647                                                    ; DATA ADDRESS VECTOR
6648                                                    .EVEN
6649  032614  001170  000000          DTA:    .WORD   $REG0,0
6650  032620  001170  001172  000000  DTB:    .WORD   $REG0,$REG1,0
6651  032626  001170  001172  001174  DTC:    .WORD   $REG0,$REG1,$REG2,$REG3,0
6652  032634  001176  000000
6653  032640  001242  002330  000000  DTD:    .WORD   $TMPS,FPS,0
6654  032646  001252  002330  000000  DTE:    .WORD   $TMP11,FPS,0
6655  032654  001256  002330  000000  DTF:    .WORD   $TMP13,FPS,0
6656  032662  001244  002332  002344  DTG:    .WORD   $TMP6,FEC,EXPFEA,FEA,0
6657  032670  002334  000000
6658  032674  001254  002332  002344  DTH:    .WORD   $TMP12,FEC,EXPFEA,FEA,0
6659  032702  002334  000000
6660  032706  001260  002332  002344  DTI:    .WORD   $TMP14,FEC,EXPFEA,FEA,0
6661  032714  002334  000000
6662  032720  001234  001236  001170  DTJ:    .WORD   $TMP2,$TMP3,$REG0,$REG1,0
6663  032726  001172  000000
6664  032732  001240  001242  001244  DTK:    .WORD   $TMP4,$TMP5,$TMP6,$TMP7
6665  032740  001246
6666  032742  001170  001172  001174          .WORD   $REG0,$REG1,$REG2,$REG3,0
6667  032750  001176  000000
6668  032754  001244  001246  001250  DTL:    .WORD   $TMP6,$TMP7,$TMP10,$TMP11
6669  032762  001252
6670  032764  001170  001172  001174          .WORD   $REG0,$REG1,$REG2,$REG3,0
6671  032772  001176  000000
6672  032776  002336  002340  002342  DTM:    .WORD   FPPOPC,FPPOPS,FPPOSP,FPS,FEC,FEA,0
6673  033004  002330  002332  002334
6674  033012  000000
6675  033014  001170  001172  002352  DTN:    .WORD   $REG0,$REG1,EREG2,0
6676  033022  000000
6677  033024  001170  001172  002354  DTO:    .WORD   $REG0,$REG1,EREG3,0
6678  033032  000000
6679  033034  001170  001172  002346  DTP:    .WORD   $REG0,$REG1,EREG0,EREG2,0
6680  033042  002352  000000
6681  033046  001170  001172  001174  DTQ:    .WORD   $REG0,$REG1,$REG2,EREG0,0
6682  033054  002346  000000
6683  033060  001170  001172  001174  DTR:    .WORD   $REG0,$REG1,$REG2,EREG2,0
6684  033066  002352  000000
6685  033072  001170  001172  001174  DTS:    .WORD   $REG0,$REG1,$REG2,EREG4,0
6686  033100  002356  000000
6687  033104  001170  001172  001174  DTT:    .WORD   $REG0,$REG1,$REG2,EREG5,0
6688  033112  002360  000000
6689  033116  001170  001172  001174  DTU:    .WORD   $REG0,$REG1,$REG2,EREG1,EREG3,0
6690  033124  002350  002354  000000
6691  033132  001170  001172  001174  DTV:    .WORD   $REG0,$REG1,$REG2,EREG3,EREG5,0
6692  033140  002354  002360  000000
6693  033146  001200  001202  001204  DTW:    .WORD   $REG4,$REG5,$REG6,$REG7,0
6694  033154  001206  000000
6695  033160  001200  001202  001204  DTX:    .WORD   $REG4,$REG5,$REG6,$REG7,EREG4,0
6696  033166  001206  002356  000000
6697  033174  001200  001202  001204  DTY:    .WORD   $REG4,$REG5,$REG6,$REG7,EREG5,0
6698  033202  001206  002360  000000
6699  033210  001200  001202  001204  DTZ:    .WORD   $REG4,$REG5,$REG6,$REG7,EREG0,EREG3,0
6700  033216  001206  002346  002354
6701  033224  000000
6702  033226  001200  001202  001204  DTAA:   .WORD   $REG4,$REG5,$REG6,$REG7,EREG1,EREG3,0
```

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

CROSS REFERENCE TABLE -- USER SYMBOLS

```
ABASE = 000000           1086
ABSDI   021632           5350#   5360
ABSDL   021576           5341    5343#
ABSDT   021552           3834    3853    3872    3891    3910    3929    3948    3967    3986    4005    4024    5335#
ABSD1   014346           3833    3838#
ABSD10  014724           3966    3971#
ABSD11  014766           3985    3990#
ABSD12  015030           4004    4009#
ABSD13  015072           4023    4028#
ABSD2   014410           3852    3857#
ABSD3   014452           3871    3876#
ABSD4   014514           3890    3895#
ABSD5   014556           3909    3914#
ABSD6   014620           3928    3933#
ABSD7   014662           3947    3952#
ABSFI   021442           5305#   5315
ABSFL   021422           5298    5300#
ABSFT   021376           3646    3663    3680    3697    3714    3731    3748    3765    3782    3799    3816    5292#
ABSF1   013710           3645    3650#
ABSF10  014176           3764    3769#
ABSF11  014230           3781    3786#
ABSF12  014262           3798    3803#
ABSF13  014314           3815    3820#
ABSF2   013742           3662    3667#
ABSF3   013774           3679    3684#
ABSF4   014026           3696    3701#
ABSF5   014060           3713    3718#
ABSF6   014112           3730    3735#
ABSF7   014144           3747    3752#
ACOM1 = 000000           1086
ACOM2 = 000000           1086
ACPUOP= 000000           1086    1101
AOOR1   003650           1503    1518#
AOOR10  004376           1690    1712#
AOOR11  004464           1719    1744#
AOOR12  004526           1751    1765#
AOOR13  004566           1772    1786#
AOOR14  004644           1793    1813#
AOOR2   003730           1525    1546#
AOOR3   004020           1553    1579#
AOOR4   004076           1586    1606#
AOOR5   004150           1613    1633#
AOOR6   004226           1640    1660#
AOOR7   004302           1667    1683#
AOOM0 = 000000           1086
AOOM1 = 000000           1086
AOOM10= 000000           1086
AOOM11= 000000           1086
AOOM12= 000000           1086
AOOM13= 000000           1086
AOOM14= 000000           1086
AOOM15= 000000           1086
AOOM2 = 000000           1086
AOOM3 = 000000           1086
AOOM4 = 000000           1086
AOOM5 = 000000           1086
```

```
ADDW6 = 000000          1086
ADDW7 = 000000          1086
ADDW8 = 000000          1086
ADDW9 = 000000          1086
ADEVCT= 000000          1086    1092
ADEVM = 000000          1086
ADRD1   007654          2768    2769    2770    2771    2791#
ADRD10  011134          3030    3031    3032    3033    3054#
ADRD11  011272          3063    3064    3065    3066    3093#
ADRD12  011432          3102    3103    3104    3105    3131#
ADRD13  011556          3140    3141    3142    3143    3163#
ADRD14  011722          3172    3173    3174    3175    3202#
ADRD2   010010          2800    2801    2802    2803    2828#
ADRD3   010164          2837    2838    2839    2840    2868#
ADRD4   010320          2877    2878    2879    2880    2905#
ADRD5   010456          2914    2915    2916    2917    2943#
ADRD6   010626          2952    2953    2954    2955    2982#
ADRD7   011000          2991    2992    2993    2994    3021#
ADRF1   012024          3211    3212    3234#
ADRF10  012760          3426    3427    3455#
ADRF11  013066          3463    3464    3486#
ADRF12  013150          3494    3495    3510#
ADRF13  013250          3518    3519    3540#
ADRF14  013340          3548    3549    3566#
ADRF2   012132          3242    3243    3266#
ADRF3   012254          3274    3275    3304#
ADRF4   012352          3312    3313    3331#
ADRF5   012446          3339    3340    3361#
ADRF6   012550          3369    3370    3388#
ADRF7   012644          3396    3397    3418#
AENV  = 000000          1086    1097
AENVM = 000000          1086    1098
AFATAL= 000000          1086    1089
ALTN  = 125252           926#   1242    1916    1918    1954    2352    2353    2369    2372    2407    2410    2868    2943
                        3163    3202    3331    3388    3685    3686    3877    3879    4083    4084    4275    4277    4498
                        4607    4699    4700    4891    4893
ALTP  = 052525           925#   1242    1898    1900    1954    2350    2353    2371    2372    2407    2410    2791    3093
                        3266    3455    3510    3668    3669    3858    3860    4066    4067    4256    4258    4481    4588
                        4682    4683    4872    4874
ALT4N = 170360           928#   2828    3054    3131    3234    3418
ALT4P = 007417           927#   2905    2982    3304    3361    3540    3566
AMADR1= 000000          1086
AMADR2= 000000          1086
AMADR3= 000000          1086
AMADR4= 000000          1086
AMAMS1= 000000          1086
AMAMS2= 000000          1086
AMAMS3= 000000          1086
AMAMS4= 000000          1086
AMSGAD= 000000          1086    1094
AMSGLG= 000000          1086    1095
AMSGTY= 000000          1086    1088
AMTYP1= 000000          1086
AMTYP2= 000000          1086
AMTYP3= 000000          1086
AMTYP4= 000000          1086
```

CROSS REFERENCE TABLE -- USER SYMBOLS

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

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

CROSS REFERENCE TABLE -- USER SYMBOLS

```
DTAG    033360          1192    6722#
DTAH    033374          1183    1184    1186    6724#
DTAI    033410          1175    1176    6726#
DTAJ    033426          1190    6729#
DTAK    033442          1177    1178    6731#
DTAL    033462          1182    6735#
DTAM    033502          1187    1188    1189    6739#
DTAN    033522          1185    6743#
DTAO    033542          1179    1180    1181    6747#
DTAP    033564          1217    1218    1219    1220    6751#
DTB     032620          1129    1130    1154    6650#
DTC     032626          1191    1194    6651#
DTD     032640          1196    6653#
DTE     032646          1197    6654#
DTF     032654          1198    6655#
DTG     032662          1200    6656#
DTH     032674          1201    6658#
DTI     032706          1202    6660#
DTJ     032720          1206    1208    1210    1212    6662#
DTK     032732          1204    1207    1209    1211    1213    6664#
DTL     032754          1205    6668#
DTM     032776          1215    6672#
DTN     033014          1151    6675#
DTO     033024          1149    1150    6677#
DTP     033034          1144    1145    6679#
DTQ     033046          1142    1143    6681#
DTR     033060          1148    6683#
DTS     033072          1146    1147    1152    6685#
DTT     033104          1136    6687#
DTU     033116          1139    1140    1141    6689#
DTV     033132          1137    1138    6691#
DTW     033146          1172    6693#
DTX     033160          1157    1158    6695#
DTY     033174          1156    1161    6697#
DTZ     033210          1162    1163    6699#
EMA     025543          1196    1197    1198    6163#
EMAA    030005          1217    6379#
EMAB    030042          1218    6384#
EMAC    030113          1219    6391#
EMAD    030157          1220    6398#
EMB     025567          1200    1201    1202    6167#
EMC     025617          1206    1207    6172#
EMD     025665          1215    6179#
EME     025752          1194    6188#
EMF     026023          1129    6195#
EMG     026060          1130    6200#
EMH     026115          1131    6205#
EMI     026154          1132    6211#
EMJ     026211          1133    6216#
EMK     026250          1134    6222#
EML     026305          1136    1138    1141    1143    1145    1147    1148    1150    1151    6227#
EMM     026375          1139    1142    1144    1146    6237#
EMN     026477          1137    1140    1149    1152    6249#
EMO     026601          1154    6261#
EMP     026657          1156    1158    1160    1163    1165    1166    1167    1169    1171    1172    6269#
EMQ     026753          1157    1162    1164    1168    6280#
```

```
EMR     027062          1159    1161    1170    1173    6293#
EMS     027171          1176    1178    1181    1182    1184    1185    1189    1190    1191    1192    6306#
EMT     027265          1177    1179    1186    1187    6317#
EMTVEC= 000030           899#   1280*   1281*
EMU     027374          1175    1180    1183    1188    6330#
EMV     027503          1204    6343#
EMV001  001354          1129#
EMV002  001362          1130#
EMV003  001370          1131#
EMV004  001376          1132#
EMV005  001404          1133#
EMV006  001412          1134#
EMV007  001420          1136#
EMV010  001426          1137#
EMV011  001434          1138#
EMV012  001442          1139#
EMV013  001450          1140#
EMV014  001456          1141#
EMV015  001464          1142#
EMV016  001472          1143#
EMV017  001500          1144#
EMV020  001506          1145#
EMV021  001514          1146#
EMV022  001522          1147#
EMV023  001530          1148#
EMV024  001536          1149#
EMV025  001544          1150#
EMV026  001552          1151#
EMV027  001560          1152#
EMV030  001566          1154#
EMV031  001574          1156#
EMV032  001602          1157#
EMV033  001610          1158#
EMV034  001616          1159#
EMV035  001624          1160#
EMV036  001632          1161#
EMV037  001640          1162#
EMV040  001646          1163#
EMV041  001654          1164#
EMV042  001662          1165#
EMV043  001670          1166#
EMV044  001676          1167#
EMV045  001704          1168#
EMV046  001712          1169#
EMV047  001720          1170#
EMV050  001726          1171#
EMV051  001734          1172#
EMV052  001742          1173#
EMV053  001750          1175#
EMV054  001756          1176#
EMV055  001764          1177#
EMV056  001772          1178#
EMV057  002000          1179#
EMV060  002006          1180#
EMV061  002014          1181#
EMV062  002022          1182#
```

```
EMV063  002030          1183#
EMV064  002036          1184#
EMV065  002044          1185#
EMV066  002052          1186#
EMV067  002060          1187#
EMV070  002066          1188#
EMV071  002074          1189#
EMV072  002102          1190#
EMV073  002110          1191#
EMV074  002116          1192#
EMV075  002124          1194#
EMV076  002132          1196#
EMV077  002140          1197#
EMV100  002146          1198#
EMV101  002154          1200#
EMV102  002162          1201#
EMV103  002170          1202#
EMV104  002176          1204#
EMV105  002204          1205#
EMV106  002212          1206#
EMV107  002220          1207#
EMV110  002226          1208#
EMV111  002234          1209#
EMV112  002242          1210#
EMV113  002250          1211#
EMV114  002256          1212#
EMV115  002264          1213#
EMV116  002272          1215#
EMV117  002300          1217#
EMV120  002306          1218#
EMV121  002314          1219#
EMV122  002322          1220#
EMW     027541          1205    6349#
EMX     027603          1208    1209    6355#
EMY     027651          1212    1213    6362#
EMZ     027737          1210    1211    6372#
EREG0   002346          1232#   5744*   6679    6681    6699    6705    6733    6749
EREG1   002350          1233#   5745*   6689    6702    6716    6722    6737
EREG2   002352          1234#   5746*   6675    6679    6683    6708    6726    6737    6749
EREG3   002354          1235#   5747*   6677    6689    6691    6699    6702    6724
EREG4   002356          1236#   5748*   6685    6695    6711    6720    6733    6741
EREG5   002360          1237#   5749*   6687    6691    6697    6720    6726    6741    6745
EREG6   002362          1238#   5750*   5751*
EREG7   002364          1239#   5752*
ERROR1= 104117           948#   1696    1744    3021    3072    3486    3524
ERRVEC= 000004           892#   1294    1295*   1306*   5686    5687*   5689*   5692*
EXPFEA  002344          1229#   5149*   5150*   5153    5224*   5225*   5228    5315*   5318    5360*   5363    5406*   5409
                        5451*   5454    5497*   5500    5542*   5545    5589*   5592    5634*   5637    6656    6658    6660
FEA     002334          1225#   5153    5228    5318    5363    5409    5454    5500    5545    5592    5637    6656    6658
                        6660    6672
FEC     002332          1224#   5142*   5151    5217*   5226    5308*   5316    5353*   5361    5399*   5407    5444*   5452
                        5490*   5498    5535*   5543    5582*   5590    5627*   5635    5661*   6656    6658    6660    6672
FPPILT  023316          1315    5657#
FPPOPC  002336          1226#   5658*   5664    6672
FPPOPS  002340          1227#   5659*   5663    6672
FPPOSP  002342          1228#   5657*   6672
```

```
FPPVEC= 000244            914#  1315*  1316*
FPS     002330           1223#  5141*  5144   5216*  5219   5307*  5310   5352*  5355   5398*  5401   5443*  5446
                         5489*  5492   5534*  5537   5581*  5584   5626*  5629   5660*  6653   6654   6655   6672
FPS0  = 000000            942#  1454   1457   1486   1489
FPS1  = 147757            941#  1438   1441   1470   1473
FIN   = 140200            938#
FIP   = 040200            937#
GNS   = ****** U          956   6102   6103   6104   6105
HOTWRM  024100           1390*  1392*  5806#
HT    = 000011            802#  5896   5937
IOTVEC= 000020            897#  1278*  1279*
LDAOL   020550           5135   5137#
LDAOT   020524           1857   1875   1893   1911   1929   1947   1965   1977   1989   2001   2013   2025   2038
                         2050   2062   2074   2086   2098   2111   2123   2135   2147   2159   2171   2184   2196
                         2208   2220   2232   2245   2257   2269   2281   2293   5129#
LDAO1   005020           1856   1861#  1964   2037   2110   2183   2244
LDAO2   005066           1874   1879#  1976   2049   2122   2195   2256
LDAO3   005134           1892   1897#  1988   2061   2134   2207   2268
LDAO4   005202           1910   1915#  2000   2073   2146   2219   2280
LDAO5   005250           1928   1933#  2012   2085   2158   2231   2292
LDAO6   005316           1946   1951#  2024   2097   2170
LDAFL   021052           5210   5212#
LDAFT   021026           2307   2326   2345   2364   2383   2402   2422   2434   2446   2458   2470   2482   2495
                         2507   2519   2531   2543   2555   2568   2580   2592   2604   2616   2628   2641   2653
                         2665   2677   2689   2702   2714   2726   2738   2750   5204#
LDAF1   006264           2306   2311#  2421   2494   2567   2640   2701
LDAF2   006336           2325   2330#  2433   2506   2579   2652   2713
LDAF3   006410           2344   2349#  2445   2518   2591   2664   2725
LDAF4   006462           2363   2368#  2457   2530   2603   2676   2737
LDAF5   006534           2382   2387#  2469   2542   2615   2688   2749
LDAF6   006606           2401   2406#  2481   2554   2627
LDARO0  020706           1855   1873   1891   1909   1927   1945   5173#
LDARO1  020722           1963   1975   1987   1999   2011   2023   5177#
LDARO2  020736           2036   2048   2060   2072   2084   2096   5181#
LDARO3  020752           2109   2121   2133   2145   2157   2169   5185#
LDARO4  020766           2182   2194   2206   2218   2230   5189#
LDARO5  021006           2243   2255   2267   2279   2291   5195#
LDARF0  021206           2305   2324   2343   2362   2381   2400   5248#
LDARF1  021230           2420   2432   2444   2456   2468   2480   5254#
LDARF2  021252           2493   2505   2517   2529   2541   2553   5260#
LDARF3  021274           2566   2578   2590   2602   2614   2626   5266#
LDARF4  021316           2639   2651   2663   2675   2687   5272#
LDARF5  021346           2700   2712   2724   2736   2748   5281#
LF    = 000012            803#  5931   5937
LGN   = 177777            932#  3702   3896   4100   4135   4294   4334   4716   4717   4910   4912
LGP   = 077777            931#  3703   3736   3737   3898   3934   3936   4101   4134   4296   4332   4750   4751
                         4948   4950
MED   = 076600            906#  1323   1371   1375   1385   5068   5076   5082   6134
MO    = 100000            930#  1952   2390   2391   2409   3787   3804   3991   4010   4185   4202   4389   4408
                         4515   4626   4801   4802   4818   4819   5005   5007   5024   5026
M1    = 177777            929#  1242   1880   1882   1934   1936   1954   2314   2315   2331   2334   2388   2391
                         2407   2410   3702   3703   3736   3737   3770   3821   3896   3898   3934   3936   3972
                         4029   4100   4101   4134   4135   4168   4219   4294   4296   4332   4334   4370   4427
                         4464   4532   4569   4645   4716   4717   4750   4751   4784   4785   4835   4836   4910
                         4912   4948   4950   4986   4988   5043   5045
NEGDI   022216          5441#  5451
```

```
NEGDL   022162          5432    5434#
NEGDT   022136          4232    4251    4270    4289    4308    4327    4346    4365    4384    4403    4422    5426#
NEGD1   015572          4231    4236#
NEGD10  016150          4364    4369#
NEGD11  016212          4383    4388#
NEGD12  016254          4402    4407#
NEGD13  016316          4421    4426#
NEGD2   015634          4250    4255#
NEGD3   015676          4269    4274#
NEGD4   015740          4288    4293#
NEGD5   016002          4307    4312#
NEGD6   016044          4326    4331#
NEGD7   016106          4345    4350#
NEGFI   022026          5396#   5406
NEGFL   022006          5389    5391#
NEGFT   021762          4044    4061    4078    4095    4112    4129    4146    4163    4180    4197    4214    5383#
NEGF1   015134          4043    4048#
NEGF10  015422          4162    4167#
NEGF11  015454          4179    4184#
NEGF12  015506          4196    4201#
NEGF13  015540          4213    4218#
NEGF2   015166          4060    4065#
NEGF3   015220          4077    4082#
NEGF4   015252          4094    4099#
NEGF5   015304          4111    4116#
NEGF6   015336          4128    4133#
NEGF7   015370          4145    4150#
NEMPAS  003142          1327    1329    1354#   5124
NMPAS1  002475          1259#   1359
OFCC    004760          1826    1847#
PIRQ    = 177772         809#
PIRQVE= 000240           903#
PREVAC  002366          1242#   5173    5177    5181    5185    5189    5195
PR0     = 000000         826#
PR1     = 000040         827#
PR2     = 000100         828#
PR3     = 000140         829#
PR4     = 000200         830#
PR5     = 000240         831#
PR6     = 000300         832#
PR7     = 000340         833#
PS      = 177776         806#    807
PSW     = 177776         807#    1838
PWRVEC= 000024           898#    1284*   1285*   6112*   6113*   6122*   6128*   6142*   6143*
RESVEC= 000010           893#
RTLAG   = 000144         910#    1375    5076
RUNAMI= 000022           908#    1323    1371    5068
SMN     = 100200         934#    3753    3953    4118    4151    4315    4351    4767    4768    4967    4969
SMP     = 000200         933#    3719    3720    3754    3915    3917    3955    4117    4152    4313    4353    4733    4734
                         4929    4931
STACK   = 001100         797#    1276    1354
START   002506           960     1269#   6147
STKLMT= 177774           808#
SUBPAS  003200          1357    1371#   5086
SWR     001146          1024#   1274    1296*   1298    1304*   1311*   1356    1377    1380    5072    5111    5681    5695
                        5703    5710    5756    5763    5775    5778    6120    6135*
```

```
SWREG   000176            958#   1304
SW0   = 000001            861#
SW00  = 000001            851#    861    1380
SW01  = 000002            850#    860    1377    5072
SW02  = 000004            849#    859
SW03  = 000010            848#    858
SW04  = 000020            847#    857
SW05  = 000040            846#    856
SW06  = 000100            845#    855
SW07  = 000200            844#    854
SW08  = 000400            843#    853
SW09  = 001000            842#    852
SW1   = 000002            860#
SW10  = 002000            841#   5111
SW11  = 004000            840#
SW12  = 010000            839#
SW13  = 020000            838#
SW14  = 040000            837#
SW15  = 100000            836#
SW2   = 000004            859#
SW3   = 000010            858#
SW4   = 000020            857#
SW5   = 000040            856#
SW6   = 000100            855#
SW7   = 000200            854#
SW8   = 000400            853#
SW9   = 001000            852#
TBITVE= 000014            894#
TKVEC = 000060            901#
TPVEC = 000064            902#
TRAPVE= 000034            900#   1282*  1283*
TRTVEC= 000014            895#
TSTDI   023166           5624#   5634
TSTDL   023132           5615    5617#
TSTDT   023106           4848    4867    4886  4905  4924  4943  4962  4981  5000  5019  5038  5609#
TSTD1   017566           4847    4852#
TSTD10  020144           4980    4985#
TSTD11  020206           4999    5004#
TSTD12  020250           5018    5023#
TSTD13  020312           5037    5042#
TSTD2   017630           4866    4871#
TSTD3   017672           4885    4830#
TSTD4   017734           4904    4909#
TSTD5   017776           4923    4928#
TSTD6   020040           4942    4947#
TSTD7   020102           4961    4966#
TSTFI   022776           5579#   5589
TSTFL   022756           5572    5574#
TSTFT   022732           4660    4677    4694  4711  4728  4745  4762  4779  4796  4813  4830  5566#
TSTF1   017130           4659    4664#
TSTF10  017416           4778    4783#
TSTF11  017450           4795    4800#
TSTF12  017502           4812    4817#
TSTF13  017534           4829    4834#
TSTF2   017162           4676    4681#
TSTF3   017214           4693    4698#
```

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

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

```
TST5    003530          1469#
TST50   005704          2125    2132#
TST51   005724          2137    2144#
TST52   005744          2149    2156#
TST53   005764          2161    2168#
TST54   006004          2173    2181#
TST55   006024          2186    2193#
TST56   006044          2198    2205#
TST57   006064          2210    2217#
TST6    003556          1485#
TST60   006104          2222    2229#
TST61   006124          2234    2242#
TST62   006144          2247    2254#
TST63   006164          2259    2266#
TST64   006204          2271    2278#
TST65   006224          2283    2290#
TST66   006244          2295    2304#
TST67   006316          2309    2323#
TST7    003604          1502#
TST70   006370          2328    2342#
TST71   006442          2347    2361#
TST72   006514          2366    2380#
TST73   006566          2385    2399#
TST74   006640          2404    2419#
TST75   006660          2424    2431#
TST76   006700          2436    2443#
TST77   006720          2448    2455#
TYPE  = 104401          1318    1326    1328    1359    1364    5113    5758    5766    5805    5824    5826    5827    5830
                        5832    5835    5838    5845    5849    5901    6054    6102#   6144
TYPOC = 104402          5814    5834    5837    5842    6103#
TYPON = 104404          6105#
TYPOS = 104403          1362    6104#
WFLAG = 000344           911#   1385    5082
ZX1MN = 100177           936#   1934    1936    3821    4029    4219    4427    4835    4836    5043    5045
ZX1MP = 000177           935#   3770    3972    4168    4370    4532    4645    4784    4785    4986    4988
$APTHD  001000           982     988#
$ASTAT= ****** U        5971    5986
$ATYC   024620          5942    5944#
$ATY1   024574          5940#
$ATY3   024602          5886    5941#
$ATY4   024612          5771    5943#
$AUTOB  001142          1021#
$BDADR  001130          1016#
$BDDAT  001134          1018#
$BELL   001314          1078#   5113    5758    5790
$CHARC  024570          5903#   5913#   5920    5929#   5934#
$CKSUM= ****** U        6108
$CMTAG  001100          1003#   1271    1272    1280    1286    1287    1288
$CM1  = 000020          1036#   1037#   1038#   1039#   1040#   1041#   1042#   1043#   1044#   1045#   1046#   1047#   1048#
                        1049#   1050#   1051#   1052#
$CM2  = 000040          1036#   1037#   1038#   1039#   1040#   1041#   1042#   1043#   1044#   1045#   1046#   1047#   1048#
                        1049#   1050#   1051#   1052#
$CM3  = 000020          1034#   1036
$CM4  = 000030          1052#   1053#   1054#   1055#   1056#   1057#   1058#   1059#   1060#   1061#   1062#   1063#   1064#
                        1065#   1066#   1067#   1068#   1069#   1070#   1071#   1072#   1073#   1074#   1075#   1076#
$CPUOP  001352          1101#
```

```
$CRLF   00132:          1080#  1364   5766   5790   5806   5826   5832   5849   5902   5937
$CTR  = 000006          1959#  1971#  1983#  1995#  2007#  2019#  2031#  2032#  2044#  2056#  2068#  2080#  2092#
                        2104#  2105#  2117#  2129#  2141#  2153#  2165#  2177#  2178#  2190#  2202#  2214#  2226#
                        2238#  2239#  2251#  2263#  2275#  2287#  2299#  2416#  2428#  2440#  2452#  2464#  2476#
                        2488#  2489#  2501#  2513#  2525#  2537#  2549#  2561#  2562#  2574#  2586#  2598#  2610#
                        2622#  2634#  2635#  2647#  2659#  2671#  2683#  2695#  2696#  2708#  2720#  2732#  2744#
                        2756#
$DEVCT  001334          1092#
$DOAGN  020520          5107   5116   5124#
$ENDAD  020510           968   5119#  5785
$ENDCT  020460          1286   5109#
$ENV    001344          1097#  5768   5881   5949   5973
$ENVM   001345          1098#  1309   5883   5888   5951
$EOP    020422          5070   5073   5085   5099#
$EOPCT  020452          1286#  5106#  5110
$ERFLG  001104          1006#  5100#  5699   5701   5707#  5729   5753#  5790
$ERMAX  001122          1013#  1289#  5701   5724#  5729
$ERROR  023634          1280   5743#
$ERRPC  001124          1014#  5760#  5761#  5762   5790   5813   5852
$ERRTB  001354          1112#  5821
$ERTTL  001116          1011#  5759#  5790
$ESCAP  001312          1077#  1288#  5723#  5781   5783   5790
$ETABL  001344          1096#
$ETEND  001354           994   1108#
$FATAL  001326          1089#  5977#
$FFLG   025040          5940#  5943#  5971   5980#  5988#
$FILLC  001164          1032#  5906   5937
$FILLS  001163          1031#  5937
$GOADR  001126          1015#
$GODAT  001132          1017#
$GET42  020500          5112   5115#
$GTSWR= ****** U        6107
$HD   = 000000           775
$HIBTS  001000           989#
$ICNT   001106          1007#  5714#  5715   5717#  5728
$ILLUP  025510          6112   6128   6149#
$INTAG  001143          1022#
$ITEMB  001120          1012#  5762#  5770   5790   5810
$LF     001322          1081#  5790   5937
$LFLG   025037          5981#  5987#
$LPROR  001110          1008#  1290#  5705#  5721#  5726   5728
$LPERR  001114          1010#  1291#  1399#  1421#  1827#  5135#  5210#  5298#  5341#  5389#  5432#  5480#  5523#
                        5572#  5615#  5705   5722#  5728   5780
$LPTST  001112          1009#  5697
$MAIL   001324           990    994   1087#  1308   5720   5768   5881
$MBADR  001002           990#
$MFLG   025036          5941#  5947   5982#  5986#
$MSGAD  001340          1094#  5957#  5960
$MSGLG  001342          1095#  5962#
$MSGTY  001324          1088#  5955   5963#  5975   5979#
$MXCNT  023632          5718   5728#
$NULL   001162          1030#  5908   5937
$NWTST= 000001          1394#  1416#  1434#  1450#  1466#  1482#  1499#  1521#  1549#  1582#  1609#  1636#  1663#
                        1686#  1715#  1747#  1768#  1789#  1821#  1851#  1869#  1887#  1905#  1923#  1941#  1959#
                        1971#  1983#  1995#  2007#  2019#  2032#  2044#  2056#  2068#  2080#  2092#  2105#  2117#
                        2129#  2141#  2153#  2165#  2178#  2190#  2202#  2214#  2226#  2239#  2251#  2263#  2275#
```

```
                        3062   3101   3139   3171   3210   3241   3273   3311   3338   3368   3395   3425   3462
                        3493   3517   3547   3578   3645   3662   3679   3696   3713   3730   3747   3764   3781
                        3798   3815   3833   3852   3871   3890   3909   3928   3947   3966   3985   4004   4023
                        4043   4060   4077   4094   4111   4128   4145   4162   4179   4196   4213   4231   4250
                        4269   4288   4307   4326   4345   4364   4383   4402   4421   4441   4458   4475   4492
                        4509   4526   4544   4563   4582   4601   4620   4639   4659   4676   4693   4710   4727
                        4744   4761   4778   4795   4812   4829   4847   4866   4885   4904   4923   4942   4961
                        4980   4999   5018   5037   5671   5672   5673   5674   5675   5681   5693   5695   5696
                        5699   5700   5701   5708   5709   5710   5722   5725   5728   5735   5736   5737   5738
                        5739   5756   5763   5775   5778   5790   6148
$SWREG  001346          1099#  1311
$SWRMK= 000000          5675   5697
$TESTN  001330          1090#  5720*  5851
$TIMES  001310          1076#  1287*  5102*  5708*  5715   5718*  5728
$TKB    001154          1027#
$TKS    001152          1026#
$TMP0   001230          1052#  5132   5207   5295   5338   5386   5429   5477   5520   5569   5612
$TMP1   001232          1053#
$TMP10  001250          1060#  6668
$TMP11  001252          1061#  6654   6668
$TMP12  001254          1062#  6658
$TMP13  001256          1063#  6655
$TMP14  001260          1064#  6660
$TMP15  001262          1065#
$TMP16  001264          1066#
$TMP17  001266          1067#
$TMP2   001234          1054#  6662
$TMP20  001270          1068#
$TMP21  001272          1069#
$TMP22  001274          1070#
$TMP23  001276          1071#
$TMP24  001300          1072#
$TMP25  001302          1073#
$TMP26  001304          1074#
$TMP27  001306          1075#
$TMP3   001236          1055#  6662
$TMP4   001240          1056#  6664
$TMP5   001242          1057#  6653   6664
$TMP6   001244          1058#  6656   6664   6668
$TMP7   001246          1059#  6664   6668
$TN   = 000277           764#   775   1394   1398*  1416   1420*  1434   1438*  1450   1454*  1466   1470*  1482
                        1486#  1499   1503#  1516   1521   1525#  1544   1549   1553#  1577   1582   1586#  1604
                        1609   1613#  1631   1636   1640#  1658   1663   1667#  1681   1686   1690#  1710   1715
                        1719#  1742   1747   1751#  1763   1768   1772#  1784   1789   1793#  1811   1821   1825#
                        1845   1851   1855#  1859   1869   1873#  1877   1887   1891#  1895   1905   1909#  1913
                        1923   1927#  1931   1941   1945#  1949   1959   1963#  1967   1971   1975#  1979   1983
                        1987#  1991   1995   1999#  2003   2007   2011#  2015   2019   2023#  2027   2032   2036#
                        2040   2044   2048#  2052   2056   2060#  2064   2068   2072#  2076   2080   2084#  2088
                        2092   2096#  2100   2105   2109#  2113   2117   2121#  2125   2129   2133#  2137   2141
                        2145#  2149   2153   2157#  2161   2165   2169#  2173   2178   2182#  2186   2190   2194#
                        2198   2202   2206#  2210   2214   2218#  2222   2226   2230#  2234   2239   2243#  2247
                        2251   2255#  2259   2263   2267#  2271   2275   2279#  2283   2287   2291#  2295   2301
                        2305#  2309   2320   2324#  2328   2339   2343#  2347   2358   2362#  2366   2377   2381#
                        2385   2396   2400#  2404   2416   2420#  2424   2428   2432#  2436   2440   2444#  2448
                        2452   2456#  2460   2464   2468#  2472   2476   2480#  2484   2489   2493#  2497   2501
                        2505#  2509   2513   2517#  2521   2525   2529#  2533   2537   2541#  2545   2549   2553#
```

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

$TPB    001160          1029#   5926*   5937
$TPFLG  001165          1033#   5875    5937
$TPS    001156          1028#   5924    5937
$TRAP   025270          1282    6079#
$TRAP2  025312          6090#   6101
$TRP  = 000005          6094#   6103#   6104#   6105#   6106#
$TRPAD  025324          6084    6101#
$TSTM   001004           991#
$TSTNM  001102          1005#   1393*   5101*   5697    5719*   5720    5725    5729    5755    5790
$TYPBN= ****** U        6106
$TYPDS= ****** U        6106
$TYPE   024312          5875#   5968    6094    6102
$TYPEC  024524          5905    5912    5919    5924#   5925
$TYPER  024076          5765    5804#
$TYPEX  024572          5930    5932    5935#
$TYPOC  025066          6024#   6103
$TYPON  025102          6023    6026#   6105
$TYPOS  025042          6019#   6104
$UNIT   001336          1093#
$UNITM  001010           993#
$USWR   001350          1100#
$XTSTR  023366          5684#
$OFILL  025265          6020*   6024*   6034    6069#
$40CAT= ****** U        5681    5765
.     = 033566           952#    956#    966     967#    969#    971#    972#    978     979#    981#    983#    1002#   1082
                        1275    1290    1291    5134    5209    5297    5340    5388    5431    5479    5522    5571    5614
                        5728    5729    5790    5857#   5937    5989#   6124    6150
.$ASTA= ****** U        5941    5944
```

```
.$X   = 001000            978#    983
```

```
PUSH      304#  5944  5946  5967  6114  6120
REPORT    304#
SBTST1    764#  3640  3657  3674  3691  3708  3725  3742  3759  3776  3793  3810  3828  3847  3866
          3885  3904  3923  3942  3961  3980  3999  4018  4038  4055  4072  4089  4106  4123  4140
          4157  4174  4191  4208  4226  4245  4264  4283  4302  4321  4340  4359  4378  4397  4416
          4436  4453  4470  4487  4504  4521  4539  4558  4577  4596  4615  4634  4654  4671  4688
          4705  4722  4739  4756  4773  4790  4807  4824  4842  4861  4880  4899  4918  4937  4956
          4975  4994  5013  5032
SBTST2    764#  1855  1873  1891  1909  1927  1945  1963  1975  1987  1999  2011  2023  2036  2048
          2060  2072  2084  2096  2109  2121  2133  2145  2157  2169  2182  2194  2206  2218  2230
          2243  2255  2267  2279  2291  2305  2324  2343  2362  2381  2400  2420  2432  2444  2456
          2468  2480  2493  2505  2517  2529  2541  2553  2566  2578  2590  2602  2614  2626  2639
          2651  2663  2675  2687  2700  2712  2724  2736  2748
SCOM0     764#
SCOM1     764#
SCOM2     764#
SCOM3     764#
SCOM4     764#
SCOM5     764#
SCOM6     764#
SCOM7     764#
SCOPE     799#  1397  1419  1437  1453  1469  1485  1502  1524  1552  1585  1612  1639  1666  1689
          1718  1750  1771  1792  1824  1854  1872  1890  1908  1926  1944  1962  1974  1986  1998
          2010  2022  2035  2047  2059  2071  2083  2095  2108  2120  2132  2144  2156  2168  2181
          2193  2205  2217  2229  2242  2254  2266  2278  2290  2304  2323  2342  2361  2380  2399
          2419  2431  2443  2455  2467  2479  2492  2504  2516  2528  2540  2552  2565  2577  2589
          2601  2613  2625  2638  2650  2662  2674  2686  2699  2711  2723  2735  2747  2766  2798
          2835  2875  2912  2950  2989  3028  3061  3100  3138  3170  3209  3240  3272  3310  3337
          3367  3394  3424  3461  3492  3516  3546  3577  3644  3661  3678  3695  3712  3729  3746
          3763  3780  3797  3814  3832  3851  3870  3889  3908  3927  3946  3965  3984  4003  4022
          4042  4059  4076  4093  4110  4127  4144  4161  4178  4195  4212  4230  4249  4268  4287
          4306  4325  4344  4363  4382  4401  4420  4440  4457  4474  4491  4508  4525  4543  4562
          4581  4600  4619  4638  4658  4675  4692  4709  4726  4743  4760  4777  4794  4811  4828
          4846  4865  4884  4903  4922  4941  4960  4979  4998  5017  5036  5056
SCPLUR    764#  5680
SEADAT    764#  1866  1884  1902  1920  1938  1956  2317  2336  2355  2374  2393  2412  3653  3670
          3687  3704  3721  3738  3755  3772  3789  3806  3823  3843  3862  3881  3900  3919  3938
          3957  3976  3995  4014  4033  4051  4068  4085  4102  4119  4136  4153  4170  4187  4204
          4221  4241  4260  4279  4298  4317  4336  4355  4374  4393  4412  4431  4449  4466  4483
          4500  4517  4534  4554  4573  4592  4611  4630  4649  4667  4684  4701  4718  4735  4752
          4769  4786  4803  4820  4837  4857  4876  4895  4914  4933  4952  4971  4990  5009  5028
          5047
SETPRI    904#
SETREG    764#  5744
SETTRA    6094#  6103  6104  6105
SETUP     764#  904#  1269
SKIP      904#  1516  1544  1577  1604  1631  1658  1681  1710  1742  1763  1784  1811  1845  1859
          1877  1895  1913  1931  1949  1967  1979  1991  2003  2015  2027  2040  2052  2064  2076
          2088  2100  2113  2125  2137  2149  2161  2173  2186  2198  2210  2222  2234  2247  2259
          2271  2283  2295  2309  2328  2347  2366  2385  2404  2424  2436  2448  2460  2472  2484
          2497  2509  2521  2533  2545  2557  2570  2582  2594  2606  2618  2630  2643  2655  2667
          2679  2691  2704  2716  2728  2740  2752  2788  2825  2865  2902  2940  2979  3018  3051
          3091  3128  3160  3199  3231  3263  3301  3328  3358  3385  3415  3452  3483  3507  3538
          3563  3648  3665  3682  3699  3716  3733  3750  3767  3784  3801  3818  3836  3855  3874
          3893  3912  3931  3950  3969  3988  4007  4026  4046  4063  4080  4097  4114  4131  4148
          4165  4182  4199  4216  4234  4253  4272  4291  4310  4329  4348  4367  4386  4405  4424
```

```
            4444    4461    4478    4495    4512    4529    4547    4566    4585    4604    4623    4642    4662    4679    4696
            4713    4730    4747    4764    4781    4798    4815    4832    4850    4869    4888    4907    4926    4945    4964
            4983    5002    5021    5040
SLASH        904#   1320    1347
SPACE        904#
STARS        904#    964     975     977     984     998    1082    1085    1265    1267    1350    1352    1367    1369    1394
            1396    1416    1418    1434    1436    1450    1452    1466    1468    1482    1484    1495    1497    1499    1501
            1521    1523    1549    1551    1582    1584    1609    1611    1636    1638    1663    1665    1686    1688    1715
            1717    1747    1749    1768    1770    1789    1791    1817    1819    1821    1823    1851    1853    1869    1871
            1887    1889    1905    1907    1923    1925    1941    1943    1959    1961    1971    1973    1983    1985    1995
            1997    2007    2009    2019    2021    2032    2034    2044    2046    2056    2058    2068    2070    2080    2082
            2092    2094    2105    2107    2117    2119    2129    2131    2141    2143    2153    2155    2165    2167    2178
            2180    2190    2192    2202    2204    2214    2216    2226    2228    2239    2241    2251    2253    2263    2265
            2275    2277    2287    2289    2301    2303    2320    2323    2339    2341    2358    2360    2377    2379    2396
            2398    2416    2418    2428    2430    2440    2442    2452    2454    2464    2466    2476    2478    2489    2491
            2501    2503    2513    2515    2525    2527    2537    2539    2549    2551    2562    2564    2574    2576    2586
            2588    2598    2600    2610    2612    2622    2624    2635    2637    2647    2649    2659    2661    2671    2673
            2683    2685    2696    2698    2708    2710    2720    2722    2732    2734    2744    2746    2759    2761    2763
            2765    2795    2797    2832    2834    2872    2874    2909    2911    2947    2949    2986    2988    3025    3027
            3058    3060    3097    3099    3135    3137    3167    3169    3206    3208    3237    3239    3269    3271    3307
            3309    3334    3336    3364    3366    3391    3393    3421    3423    3458    3460    3489    3491    3513    3515
            3543    3545    3570    3572    3574    3576    3640    3643    3657    3660    3674    3677    3691    3694    3708
            3711    3725    3728    3742    3745    3759    3762    3776    3779    3793    3796    3810    3813    3828    3831
            3847    3850    3866    3869    3885    3888    3904    3907    3923    3926    3942    3945    3961    3964    3980
            3983    3999    4002    4018    4021    4038    4041    4055    4058    4072    4075    4089    4092    4106    4109
            4123    4126    4140    4143    4157    4160    4174    4177    4191    4194    4208    4211    4226    4229    4245
            4248    4264    4267    4283    4286    4302    4305    4321    4324    4340    4343    4359    4362    4378    4381
            4397    4400    4416    4419    4436    4439    4453    4456    4470    4473    4487    4490    4504    4507    4521
            4524    4539    4542    4558    4561    4577    4580    4596    4599    4615    4618    4634    4637    4654    4657
            4671    4674    4688    4691    4705    4708    4722    4725    4739    4742    4756    4759    4773    4776    4790
            4793    4807    4810    4824    4827    4842    4845    4861    4864    4880    4883    4899    4902    4918    4921
            4937    4940    4956    4959    4975    4978    4994    4997    5013    5016    5032    5035    5052    5089    5332
            5423    5514    5606    5668    5731    5790    5860    5939    5996    6073    6110    6126
STATUS       764#
SWRSU        904#   1292#
TADOD1       764#   2763    2795    2832    2872    2909    2947    2986    3025    3058    3097    3135    3167
TADOD2       764#   2777    2809    2849    2886    2924    2963    3007    3040    3112    3149    3183
TADOF1       764#   3206    3237    3269    3307    3334    3364    3391    3421    3458    3489    3513    3543
TADOF2       764#   3219    3251    3284    3321    3346    3378    3403    3435    3476    3500    3556
TADOR1       764#   1499    1521    1549    1582    1609    1636    1663    1686    1715    1747    1768    1789
TADOR2       764#   1510    1533    1561    1593    1620    1647    1675    1704    1731    1757    1778    1800
TRMTRP      6094#
TYPBIN       904#
TYPDEC       904#
TYPNAM       904#
TYPNUM       904#
TYPOCS       904#
TYPOCT       904#
TYPTXT       904#
UPCODE       764#   6133
SSCMRE       996#   1036    1037    1038    1039    1040    1041    1042    1043    1044    1045    1046    1047    1048    1049
            1050    1051
SSCMTM       996#   1052    1053    1054    1055    1056    1057    1058    1059    1060    1061    1062    1063    1064    1065
            1066    1067    1068    1069    1070    1071    1072    1073    1074    1075
SSESCA       904#
SSNEWT       904#   1394    1416    1434    1450    1466    1482    1499    1521    1549    1582    1609    1636    1663    1686
```

```
            1715    1747    1768    1789    1821    1851    1869    1887    1905    1923    1941    1959    1971    1983    1995
            2007    2019    2032    2044    2056    2068    2080    2092    2105    2117    2129    2141    2153    2165    2178
            2190    2202    2214    2226    2239    2251    2263    2275    2287    2301    2320    2339    2358    2377    2396
            2416    2428    2440    2452    2464    2476    2489    2501    2513    2525    2537    2549    2562    2574    2586
            2598    2610    2622    2635    2647    2659    2671    2683    2696    2708    2720    2732    2744    2763    2795
            2832    2872    2909    2947    2986    3025    3058    3097    3135    3167    3206    3237    3269    3307    3334
            3364    3391    3421    3458    3489    3513    3543    3574    3640    3657    3674    3691    3708    3725    3742
            3759    3776    3793    3810    3828    3847    3866    3885    3904    3923    3942    3961    3980    3999    4018
            4038    4055    4072    4089    4106    4123    4140    4157    4174    4191    4208    4226    4245    4264    4283
            4302    4321    4340    4359    4378    4397    4416    4436    4453    4470    4487    4504    4521    4539    4558
            4577    4596    4615    4634    4654    4671    4688    4705    4722    4739    4756    4773    4790    4807    4824
            4842    4861    4880    4899    4918    4937    4956    4975    4994    5013    5032
$$SET      6094#   6103    6104    6105
$$SETM     1308#
$$SKIP      904#   1516    1544    1577    1604    1631    1658    1681    1710    1742    1763    1784    1811    1845    1859
            1877    1895    1913    1931    1949    1967    1979    1991    2003    2015    2027    2040    2052    2064    2076
            2088    2100    2113    2125    2137    2149    2161    2173    2186    2198    2210    2222    2234    2247    2259
            2271    2283    2295    2309    2328    2347    2366    2385    2404    2424    2436    2448    2460    2472    2484
            2497    2509    2521    2533    2545    2557    2570    2582    2594    2606    2618    2630    2643    2655    2667
            2679    2691    2704    2716    2728    2740    2752    2789    2826    2866    2903    2941    2980    3019    3052
            3091    3129    3161    3200    3232    3264    3302    3329    3359    3386    3416    3453    3484    3508    3538
            3564    3648    3665    3682    3699    3716    3733    3750    3767    3784    3801    3818    3836    3855    3874
            3893    3912    3931    3950    3969    3988    4007    4026    4046    4063    4080    4097    4114    4131    4148
            4165    4182    4199    4216    4234    4253    4272    4291    4310    4329    4348    4367    4386    4405    4424
            4444    4461    4478    4495    4512    4529    4547    4566    4585    4604    4623    4642    4662    4679    4696
            4713    4730    4747    4764    4781    4798    4815    4832    4850    4869    4888    4907    4926    4945    4964
            4983    5002    5021    5040
.EQUAT       764#    794
.HEADE       764#    765
.SBPAS       764#   5052
.SETUP       764#   1262
.STPAS       764#   1320
.$ACT1       764#    962
.$APTB       764#   1083#
.$APTH       764#    973
.$APTY       764#   5937
.$CATC       764#    950
.$CMTA       764#    996
.$EOP        764#   5089
.$ERRO       764#   5729
.$POWE       764#   6108
.$SCOP       764#   5666
.$TRAP       764#   6071
.$TYER       764#   5790
.$TYPE       764#   5858
.$TYPO       764#   5994

. ABS.   033566      000

 ERRORS DETECTED:  0
 DEFAULT GLOBALS GENERATED:  0

 DSK:DQFPAA,DSK:DQFPAA.SEQ/SOL/CRF=DQFPAA.MEM,DQFPAA.MAC,DQFPAA.P11
 RUN-TIME: 34 33 2 SECONDS
```

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