KD11K

FLOAT PT BASIC INST SET

MD-11-DQFPA-B

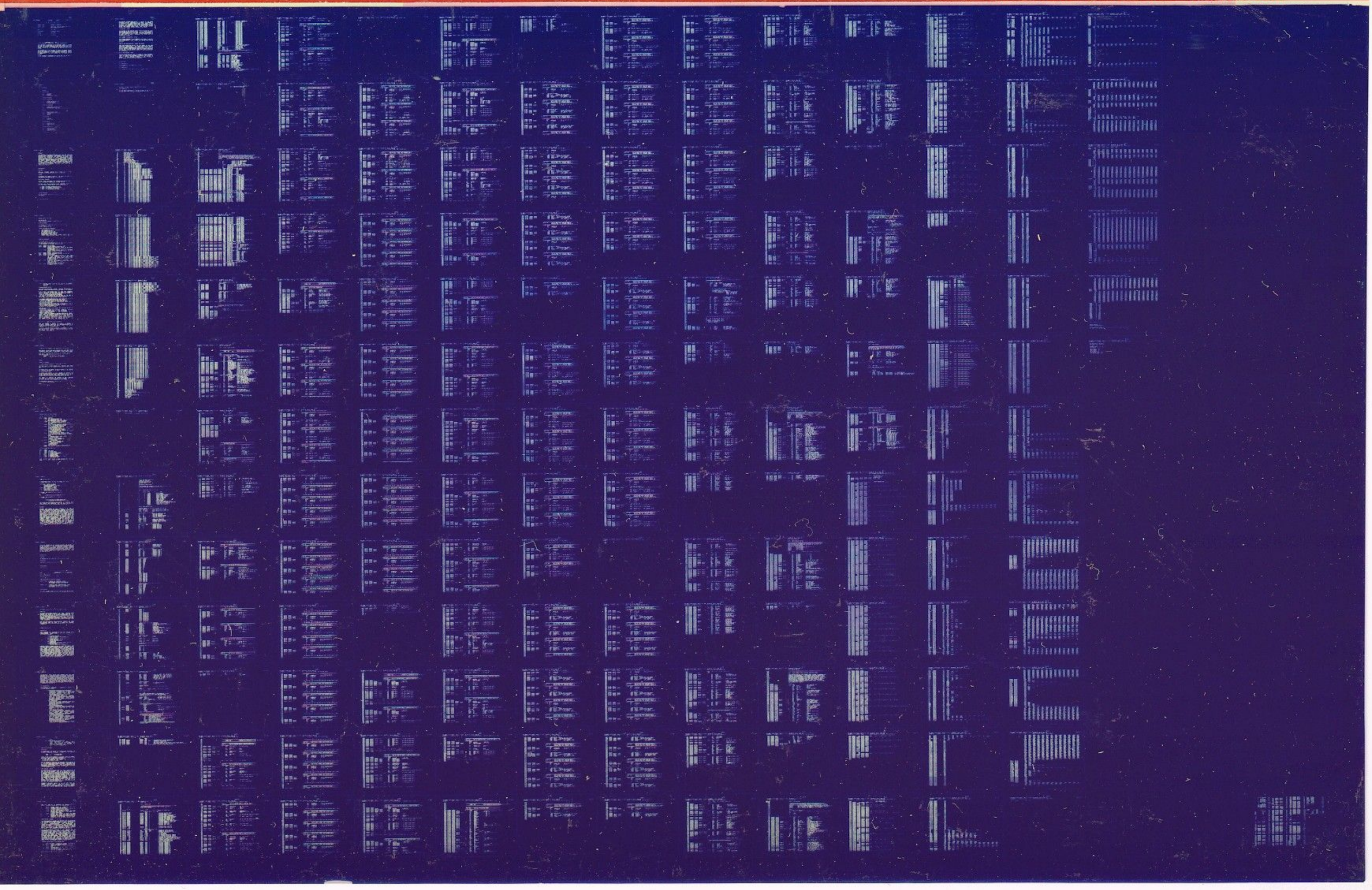
EP-DQFPA-B-DL-A

FICHE 1 OF 1

JUN 1977

digital

MADE IN USA



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

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

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

DATE : MAY, 1977

MAINTAINER: DIAGNOSTIC GROUP

AUTHOR: DON NORTH

REVISED BY: DON NORTH

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 0-34074(8). THE UPPER 2.0K WORDS ARE RESERVED FOR THE XXDP MONITOR, IF EMPLOYED.

2.3 PRELIMINARY PROGRAMS

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

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

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

3. LOADING PROCEDURE

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

4. STARTING PROCEDURE

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 "SLPTST" MATCHES THE NUMBER OF THE TEST CURRENTLY EXECUTING. AT THIS POINT, THE TEST IS LOOPED ON INDEFINATELY, UNTIL EITHER SW<8>=0 OR "SLPTST" IS CHANGED. NOTE THAT IF "SLPTST" DOES NOT MATCH THE TEST NUMBER OF ANY TEST, THE CONTENTS OF "SLPTST" ARE EFFECTIVELY IGNORED, AND EXECUTION PROCEEDS NORMALLY.

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

WHEN THE PROGRAM IS STARTED (AT 200(8)), A MESSAGE IS OPTIONALLY PRINTED INDICATING THE PRESCENCE/ABSCENCE OF AN FP11-E HOT FLOATING POINT UNIT OPTION (BASED UPON WHETHER "WWKMI" 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

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

TIMES SPECIFIED AS (MINUTES):(SECONDS)

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

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

8.2 STACK POINTER

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

THE START OF EACH TEST, AND AT THE END OF A PASS.

8.3 POWER FAIL

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

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

9. PROGRAM DESCRIPTION

9.1 ORGANIZATION

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

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

WHEREVER FEASIBLE, COMMON 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

ACO-5.

(8) SINGLE OPERAND FP INSTRUCTIONS. VERIFY FUNCTIONALITY OF 'ABS-', 'NEG-', 'TST-', AND 'CLR-' FP INSTRUCTIONS IN F/D MODES. CHECK FLOATING CC SETTINGS AND EXCEPTION CONDITIONS, WHEN APPLICABLE (-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:

- $MXCNT - 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 (STYPERR ENTRY POINT) IS CALLED BY THE ERROR PROCESSING ROUTINE DESCRIBED IN 9.3.3 ABOVE. ITS PURPOSE IS TO IMPLEMENT THE ERROR MESSAGE/DATA VALUE ERROR TYPEOUT FACILITY. THE SUBROUTINE WILL, GIVEN THE INDEXING BYTE FROM THE ERROR CALL INSTRUCTION, PICK UP THE CORRECT ERROR MESSAGE VECTOR FROM 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

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

TABLE OF CONTENTS

```
                        .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
```

```
                        ;
                        ;*GENERAL REGISTER DEFINITIONS
                        ;
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=   100000
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
```

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

                        ;*BASIC "CPU" TRAP VECTOR ADDRESSES
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          ; 0000111110000111
170360                  ALT4N=  170360          ; 1111000001111000
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=    17,777          ; 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  003000          JMP     @#START ;;JUMP TO STARTING ADDRESS OF PROGRAM

                        .SBTTL  ACT11 HOOKS

                        ;;******************************************************************
                        ;HOOKS REQUIRED BY ACT11
                        000204          $SVPC=.                 ;SAVE PC
                        000046          .=46
000046  021016          $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.
```

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

```
                                .SBTTL  COMMON TAGS

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

                                        .=1100
001100                          $CMTAG:                         ;;START OF COMMON TAGS
                                ;;-----START OF CLEAR 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                  $LPERR: .WORD   0               ;;CONTAINS SCOPE RETURN FOR ERRORS
001114  000000                  $ERTTL: .WORD   0               ;;CONTAINS TOTAL ERRORS DETECTED
001116  000000                  $ITEMB: .WORD   0               ;;CONTAINS ITEM CONTROL BYTE
001120  000001                  $ERMAX: .WORD   1               ;;CONTAINS MAX. ERRORS PER TEST
001122  000000                  $ERRPC: .WORD   0               ;;CONTAINS PC OF LAST ERROR INSTRUCTION
001124  000000                  $GDADR: .WORD   0               ;;CONTAINS ADDRESS OF 'GOOD' DATA
001126  000000                  $BDADR: .WORD   0               ;;CONTAINS ADDRESS OF 'BAD' DATA
001130  000000                  $GDDAT: .WORD   0               ;;CONTAINS 'GOOD' DATA
001132  000000                  $BDDAT: .WORD   0               ;;CONTAINS 'BAD' DATA
001134  000000                          .WORD   0               ;;RESERVED--NOT TO BE USED
001136  000000                          .WORD   0
001140  000                     $AUTOB: .BYTE   0               ;;AUTOMATIC MODE INDICATOR
001141  000                     $INTAG: .BYTE   0               ;;INTERRUPT MODE INDICATOR
001142  000000                          .WORD   0
                                ;;-----END OF CLEAR COMMON TAGS-----
001144  177570                  $SWR:   .WORD   DSWR            ;;ADDRESS OF SWITCH REGISTER
001146  177570                  DISPLA: .WORD   DDISP           ;;ADDRESS OF DISPLAY REGISTER
001150  000000                  $LPTST: .WORD   0               ;;CONTAINS TEST NUMBER TO LOOP UPON
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)
001230  000000                  $TMP0:  .WORD   0               ;;USER DEFINED
001232  000000                  $TMP1:  .WORD   0               ;;USER DEFINED
001234  000000                  $TMP2:  .WORD   0               ;;USER DEFINED
001236  000000                  $TMP3:  .WORD   0               ;;USER DEFINED
001240  000000                  $TMP4:  .WORD   0               ;;USER DEFINED
001242  000000                  $TMP5:  .WORD   0               ;;USER DEFINED
001244  000000                  $TMP6:  .WORD   0               ;;USER DEFINED
001246  000000                  $TMP7:  .WORD   0               ;;USER DEFINED
001250  000000                  $TMP10: .WORD   0               ;;USER DEFINED
001252  000000                  $TMP11: .WORD   0               ;;USER DEFINED
001254  000000                  $TMP12: .WORD   0               ;;USER DEFINED
001256  000000                  $TMP13: .WORD   0               ;;USER DEFINED
001260  000000                  $TMP14: .WORD   0               ;;USER DEFINED
001262  000000                  $TMP15: .WORD   0               ;;USER DEFINED
001264  000000                  $TMP16: .WORD   0               ;;USER DEFINED
001266  000000                  $TMP17: .WORD   0               ;;USER DEFINED
001270  000000                  $TMP20: .WORD   0               ;;USER DEFINED
001272  000000                  $TMP21: .WORD   0               ;;USER DEFINED
001274  000000                  $TMP22: .WORD   0               ;;USER DEFINED
001276  000000                  $TMP23: .WORD   0               ;;USER DEFINED
001300  000000                  $TMP24: .WORD   0               ;;USER DEFINED
001302  000000                  $TMP25: .WORD   0               ;;USER DEFINED
001304  000000                  $TMP26: .WORD   0               ;;USER DEFINED
001306  000000                  $TMP27: .WORD   0               ;;USER DEFINED
001310  000000                  $TIMES: 0                       ;;MAX. NUMBER OF ITERATIONS
001312  000000                  $ESCAPE:0                       ;;ESCAPE ON ERROR ADDRESS
001314  177607  000377            $BELL:  .ASCIZ  <207><377><377> ;;CODE FOR BELL
001320     077                  $QUES:  .ASCII  /?/             ;;QUESTION MARK
001321     015                  $CRLF:  .ASCII  <15>            ;;CARRIAGE RETURN
001322  000012                  $LF:    .ASCIZ  <12>            ;;LINE FEED
                                ;;*****************************************************************
                                .SBTTL  APT MAILBOX-ETABLE

                                ;;*****************************************************************
                                .EVEN
001324                          $MAIL:                          ;;APT MAILBOX
001324  000000                  $MSGTY: .WORD   AMSGTY          ;;MESSAGE TYPE CODE
001326  000000                  $FATAL: .WORD   AFATAL          ;;FATAL ERROR NUMBER
001330  000000                  $TESTN: .WORD   ATESTN          ;;TEST NUMBER
001332  000000                  $PASS:  .WORD   APASS           ;;PASS COUNT
001334  000000                  $DEVCT: .WORD   ADEVCT          ;;DEVICE COUNT
001336  000000                  $UNIT:  .WORD   AUNIT           ;;I/O UNIT NUMBER
001340  000000                  $MSGAD: .WORD   AMSGAD          ;;MESSAGE ADDRESS
001342  000000                  $MSGLG: .WORD   AMSGLG          ;;MESSAGE LENGTH
001344                          $ETABLE:                        ;;APT ENVIRONMENT TABLE
001344     000                  $ENV:   .BYTE   AENV            ;;ENVIRONMENT BYTE
001345     000                  $ENVM:  .BYTE   AENVM           ;;ENVIRONMENT MODE BITS
001346  000000                  $SWREG: .WORD   ASWREG          ;;APT SWITCH REGISTER
001350  000000                  $USWR:  .WORD   AUSWR           ;;USER SWITCHES
001352  000000                  $CPUOP: .WORD   ACPUOP          ;;CPU TYPE,OPTIONS
                                ;*                              BITS 15-11=CPU TYPE
                                ;*                                      11/04=01,11/05=02,11/20=03,11/40=04,11/45=05
                                ;*                                      11/70=06,PDQ=07,Q=10
                                ;*                              BIT 10=REAL TIME CLOCK
```

```
                                ;*                      BIT  9=FLOATING POINT PROCESSOR
                                ;*                      BIT  8=MEMORY MANAGEMENT
001354                          $ETEND:
                                .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  026331  030523  033126  EMV001:  .WORD   EMF,DHA,DTB      ; FPS RIPPLE 1
001362  026366  030523  033126  EMV002:  .WORD   EMG,DHA,DTB      ; FPS RIPPLE 0
001370  026423  030541  033122  EMV003:  .WORD   EMH,DHB,DTA      ; SETF
001376  026462  030541  033122  EMV004:  .WORD   EMI,DHB,DTA      ; SETD
001404  026517  030541  033122  EMV005:  .WORD   EMJ,DHB,DTA      ; SETI
001412  026556  030541  033122  EMV006:  .WORD   EMK,DHB,DTA      ; SETL
                                                ;*****  SRC/DST ADDR MODES VECTORS  *****
001420  026613  031273  033412  EMV007:  .WORD   EML,DHO,DTT      ; M05-M77 RESULT
001426  027005  031362  033440  EMV010:  .WORD   EMN,DHQ,DTV      ; M13-M35 DSTREG
001434  026613  031362  033440  EMV011:  .WORD   EML,DHQ,DTV      ; M13-M35 RESULT
001442  026703  031324  033424  EMV012:  .WORD   EMM,DHP,DTU      ; M21-M53 SRCREG
001450  027005  031324  033424  EMV013:  .WORD   EMN,DHP,DTU      ; M21-M53 DSTREG
001456  026613  031324  033424  EMV014:  .WORD   EML,DHP,DTU      ; M21-M53 RESULT, M63-M71 RESULT
001464  026703  031160  033354  EMV015:  .WORD   EMM,DHL,DTQ      ; M30-M67 SRCREG
001472  026613  031160  033354  EMV016:  .WORD   EML,DHL,DTQ      ; M30-M67 RESULT
001500  026703  031130  033342  EMV017:  .WORD   EMM,DHK,DTP      ; M42-M00 SRCREG
001506  026613  031130  033342  EMV020:  .WORD   EML,DHK,DTP      ; M42-M00 RESULT
001514  026703  031242  033400  EMV021:  .WORD   EMM,DHN,DTS      ; M54-M37 SRCREG
001522  026613  031242  033400  EMV022:  .WORD   EML,DHN,DTS      ; M54-M37 RESULT, M77-M24 RESULT
001530  026613  031211  033366  EMV023:  .WORD   EML,DHM,DTR      ; M72-M27 RESULT
001536  027005  031105  033332  EMV024:  .WORD   EMN,DHJ,DTO      ; M27-M43 DSTREG
001544  026613  031105  033332  EMV025:  .WORD   EML,DHJ,DTO      ; M27-M43 RESULT
001552  026613  031062  033322  EMV026:  .WORD   EML,DHI,DTN      ; M37-M62 RESULT, M67-M12 RESULT
001560  027005  031242  033400  EMV027:  .WORD   EMN,DHP,DTS      ; M77-M24 DSTREG
                                                ;*****  VECTOR FOR CFCC TEST  *****
001566  027107  030523  033126  EMV030:  .WORD   EMO,DHA,DTB      ; CFCC INSTR
                                                ;*****  FSRC/FDST D ADDR MODES VECTORS  *****
001574  027165  031523  033502  EMV031:  .WORD   EMP,DHT,DTY      ; M15-M67 RESULT, M67-M25 RESULT
001602  027261  031457  033466  EMV032:  .WORD   EMQ,DHS,DTX      ; M44-M37 SRCREG
001610  027165  031457  033466  EMV033:  .WORD   EMP,DHS,DTX      ; M44-M37 RESULT
001616  027370  032174  033644  EMV034:  .WORD   EMR,DHAA,DTAF    ; M75-M34 DSTREG
001624  027165  032174  033644  EMV035:  .WORD   EMP,DHAA,DTAF    ; M75-M34 RESULT
001632  027370  031523  033502  EMV036:  .WORD   EMR,DHT,DTY      ; M67-M25 DSTREG
001640  027261  031567  033516  EMV037:  .WORD   EMQ,DHU,DTZ      ; M20-M13 SRCREG
001646  027165  031567  033516  EMV040:  .WORD   EMP,DHU,DTZ      ; M20-M13 RESULT
001654  027261  032112  033624  EMV041:  .WORD   EMQ,DHZ,DTAE     ; M51-M77 SRCREG
001662  027165  032112  033624  EMV042:  .WORD   EMP,DHZ,DTAE     ; M51-M77 RESULT
```

```
405  001670  027165  031711  033552  EMV043: .WORD   EMP,DHW,DTAB     ; M27-M70 RESULT
406  001676  027165  032037  033606  EMV044: .WORD   EMP,DHY,DTAD     ; M77-M64 RESULT
407  001704  027261  031764  033570  EMV045: .WORD   EMQ,DHX,DTAC     ; M32-M27 SRCREG
408  001712  027165  031764  033570  EMV046: .WORD   EMP,DHX,DTAC     ; M32-M27 RESULT, M00-M52 RESULT
409  001720  027370  031640  033534  EMV047: .WORD   EMR,DHV,DTAA     ; M63-M41 DSTREG
410  001726  027165  031640  033534  EMV050: .WORD   EMP,DHV,DTAA     ; M63-M41 RESULT
411  001734  027165  031420  033454  EMV051: .WORD   EMP,DHR,DTW      ; M37-M03 RESULT
412  001742  027370  031764  033570  EMV052: .WORD   EMR,DHX,DTAC     ; M00-M52 DSTREG
413                                   ;***** FSRC/FDST F ADDR MODES VECTORS  *****
414  001750  027702  032371  033716  EMV053: .WORD   EMU,DHAO,DTAI    ; M12-M45 DSTREG
415  001756  027477  032371  033716  EMV054: .WORD   EMS,DHAO,DTAI    ; M12-M45 RESULT
416  001764  027573  032505  033750  EMV055: .WORD   EMT,DHAF,DTAK    ; M34-M60 SRCREG
417  001772  027477  032505  033750  EMV056: .WORD   EMS,DHAF,DTAK    ; M34-M60 RESULT
418  002000  027573  032776  034050  EMV057: .WORD   EMT,DHAJ,DTAO    ; M50-M32 SRCREG
419  002006  027702  032776  034050  EMV060: .WORD   EMU,DHAJ,DTAO    ; M50-M32 DSTREG
420  002014  027477  032776  034050  EMV061: .WORD   EMS,DHAJ,DTAO    ; M50-M32 RESULT
421  002022  027477  032563  033770  EMV062: .WORD   EMS,DHAG,DTAL    ; M72-M11 RESULT
422  002030  027702  032326  033702  EMV063: .WORD   EMU,DHAC,DTAH    ; M37-M23 DSTREG
423  002036  027477  032326  033702  EMV064: .WORD   EMS,DHAC,DTAH    ; M37-M23 RESULT, M23-M37 RESULT
424  002044  027477  032717  034030  EMV065: .WORD   EMS,DHAI,DTAN    ; M77-M75 RESULT
425  002052  027573  032326  033702  EMV066: .WORD   EMT,DHAC,DTAH    ; M23-M37 SRCREG
426  002060  027573  032641  034010  EMV067: .WORD   EMT,DHAH,DTAM    ; M45-M54 SRCREG
427  002066  027702  032641  034010  EMV070: .WORD   EMU,DHAH,DTAM    ; M45-M54 DSTREG
428  002074  027477  032641  034010  EMV071: .WORD   EMS,DHAH,DTAM    ; M45-M54 RESULT
429  002102  027477  032441  033734  EMV072: .WORD   EMS,DHAE,DTAJ    ; M27-M77 RESULT
430  002110  027477  033062  033134  EMV073: .WORD   EMS,DHAK,DTC     ; M67-M67 RESULT, M02-M27 RESULT
431  002116  027477  032263  033666  EMV074: .WORD   EMS,DHAB,DTAG    ; M61-M01 RESULT
432                                   ;***** FP ACC ALL THERE ? VECTOR  *****
433  002124  026260  030550  033134  EMV075: .WORD   EME,DHC,DTC      ; FP AC ALL THERE  ?
434                                   ;***** FPS ERROR VECTORS  *****
435  002132  026051  030606  033146  EMV076: .WORD   EMA,DHD,DTD      ; FPS - ABS,NEG,CLR,TST F
436  002140  026051  030606  033154  EMV077: .WORD   EMA,DHD,DTE      ; FPS - LDD/STD, ABS,NEG,CLR,TST D
437  002146  026051  030606  033162  EMV100: .WORD   EMA,DHD,DTF      ; FPS - LDD/LDF/STD
438                                   ;***** FEC/FEA ERROR VECTORS  *****
439  002154  026075  030622  033170  EMV101: .WORD   EMB,DHE,DTG      ; FEC/FEA - ABS,NEG,CLR,TST F
440  002162  026075  030622  033202  EMV102: .WORD   EMB,DHE,DTH      ; FEC/FEA - LDD/STD, ABS,....,CLR D
441  002170  026075  030622  033214  EMV103: .WORD   EMB,DHE,DTI      ; FEC/FEA - LDD/LDF/STD
442                                   ;***** RESULT VECTORS  *****
443  002176  030011  030720  033240  EMV104: .WORD   EMV,DHG,DTK      ; RESULT - LDD/STD
444  002204  030047  030720  033262  EMV105: .WORD   EMW,DHG,DTL      ; RESULT - LDD/LDF/STD
445  002212  026125  030662  033226  EMV106: .WORD   EMC,DHF,DTJ      ; RESULT - ABSF
446  002220  026125  030720  033240  EMV107: .WORD   EMC,DHG,DTK      ; RESULT - ABSD
447  002226  030111  030662  033226  EMV110: .WORD   EMX,DHF,DTJ      ; RESULT - NEGF
448  002234  030111  030720  033240  EMV111: .WORD   EMX,DHG,DTK      ; RESULT - NEGD
449  002242  030245  030662  033226  EMV112: .WORD   EMZ,DHF,DTJ      ; RESULT - CLRF
450  002250  030245  030720  033240  EMV113: .WORD   EMZ,DHG,DTK      ; RESULT - CLRD
451  002256  030157  030662  033226  EMV114: .WORD   EMY,DHF,DTJ      ; RESULT - TSTF
452  002264  030157  030720  033240  EMV115: .WORD   EMY,DHG,DTK      ; RESULT - TSTD
453                                   ;***** ILLEGAL TRAP CATCHER VECTOR  *****
454  002272  026173  031016  033304  EMV116: .WORD   EMD,DHH,DTM      ; ILLEGAL TRAP CATCHER
455                                   ;***** PC MODE 2 WRONG INCRE VECTORS *****
456  002300  030313  033120  034072  EMV117: .WORD   EMAA,DHAL,DTAP   ;+0
457  002306  030350  033120  034072  EMV120: .WORD   EMAB,DHAL,DTAP   ;+4
458  002314  030421  033120  034072  EMV121: .WORD   EMAC,DHAL,DTAP   ;+6
459  002322  030465  033120  034072  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  000000                  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  <CR><LF><LF><LF>"MD-11-DQFPA-"
002404  030455  026461  050504
002412  050106  026501
002416  027102                          .ASCII  "B."
002420  027056                          .ASCII  ".."
002422  042120  026520  030461          .ASCIZ  "PDP-11/6X F.P.U. BASIC INSTRUCTION TESTS"<CR><LF>
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  <CR><LF>"PASS #"
002502  020123  000043
```

```
                                .SBTTL  START OF PASS ROUTINE


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

003000                          START:
                                .SBTTL  INITIALIZE THE COMMON TAGS
                                ;;CLEAR THE COMMON TAGS ($CMTAG) AREA
003000  012706  001100                  MOV     #$CMTAG,R6      ;;FIRST LOCATION TO BE CLEARED
003004  005026                          CLR     (R6)+           ;;CLEAR MEMORY LOCATION
003006  022706  001144                  CMP     #SWR,R6 ;;DONE?
003012  001374                          BNE     .-6             ;;LOOP BACK IF NO
003014  012706  001100                  MOV     #STACK,SP       ;;SETUP THE STACK POINTER
                                ;;INITIALIZE A FEW VECTORS
003020  012737  023664  000020          MOV     #$SCOPE,@#IOTVEC ;;IOT VECTOR FOR SCOPE ROUTINE
003026  012737  000340  000022          MOV     #340,@#IOTVEC+2 ;;LEVEL 7
003034  012737  024142  000030          MOV     #$ERROR,@#EMTVEC ;;EMT VECTOR FOR ERROR ROUTINE
003042  012737  000340  000032          MOV     #340,@#EMTVEC+2 ;;LEVEL 7
003050  012737  025576  000034          MOV     #$TRAP,@#TRAPVEC ;;TRAP VECTOR FOR TRAP CALLS
003056  012737  000340  000036          MOV     #340,@#TRAPVEC+2;LEVEL 7
003064  012737  025644  000024          MOV     #$PWRON,@#PWRVEC ;;POWER FAILURE VECTOR
003072  012737  000340  000026          MOV     #340,@#PWRVEC+2 ;;LEVEL 7
003100  013737  020766  020760          MOV     $ENDCT,$EOPCT   ;;SETUP END-OF-PROGRAM COUNTER
003106  005037  001310                  CLR     $TIMES          ;;INITIALIZE NUMBER OF ITERATIONS
003112  005037  001312                  CLR     $ESCAPE         ;;CLEAR THE ESCAPE ON ERROR ADDRESS
003116  012737  000001  001120          MOV     #1,$ERMAX       ;;ALLOW ONE ERROR PER TEST
003124  012737  003124  001110          MOV     #.,$LPADR       ;;INITIALIZE THE LOOP ADDRESS FOR SCOPE
003132  012737  003132  001112          MOV     #.,$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.
003140  013746  000004                  MOV     @#ERRVEC,-(SP)  ;;SAVE ERROR VECTOR
003144  012737  003200  000004          MOV     #64$,@#ERRVEC   ;;SET UP ERROR VECTOR
003152  012737  177570  001144          MOV     #DSWR,SWR       ;;SETUP FOR A HARDWARE SWICH REGISTER
003160  012737  177570  001146          MOV     #DDISP,DISPLAY  ;;AND A HARDWARE DISPLAY REGISTER
003166  022777  177777  175750          CMP     #-1,@SWR        ;;TRY TO REFERENCE HARDWARE SWR
003174  001012                          BNE     66$             ;;BRANCH IF NO TIMEOUT TRAP OCCURRED
                                                                ;;AND  THE HARDWARE SWR IS NOT = -1
003176  000403                          BR      65$             ;;BRANCH IF NO TIMEOUT
003200  012716  003206          64$:    MOV     #65$,(SP)       ;;SET UP FOR TRAP RETURN
003204  000002                          RTI
003206  012737  000176  001144  65$:    MOV     #SWREG,SWR      ;;POINT TO SOFTWARE SWR
003214  012737  000174  001146          MOV     #DISPREG,DISPLAY
003222  012637  000004          66$:    MOV     (SP)+,@#ERRVEC  ;;RESTORE ERROR VECTOR

003226  005037  001332                  CLR     $PASS           ;;CLEAR PASS COUNT
003232  132737  000200  001345          BITB    #APTSIZE,$ENVM  ;;TEST USER SIZE UNDER APT
003240  001403                          BEQ     67$             ;;YES,USE NON-APT SWITCH
003242  012737  001346  001144          MOV     #$SWREG,SWR     ;;NO,USE APT SWITCH REGISTER
003250                          67$:

                                ; SET UP FPP UNEXPECTED TRAP CATCHER - - - - - - - - -
003250  012737  023624  000244          MOV     #FPPILT,@#FPPVEC        ; NEW PC AT FPP TRAP
003256  005037  000246                  CLR     @#FPPVEC+2              ; NEW PS AT FPP TRAP
```

```
003262  104401  002376                  TYPE    ,BGNMES         ; ID MESSAGE AT START

                                        ;////////////////////////////////////////////////////////////////////
                                        ;       MESSAGE ON WHETHER OR NOT HFP UNIT IS PRESENT

003266  076600  000022                  MED     ,RWHAMI                 ;WHAMI INTO R0
003272  032700  000020                  BIT     #BIT04,R0               ;IS THERE A HFP UNIT ?
003276  001403                          BEQ     70$                     ;NO, BR
003300  104401  003314                  TYPE    ,68$                    ;INDICATE FP11-E PRESENT
003304  000453                          BR      NEWPAS                  ;GO FOR SUBPASS INIT
003306  104401  003354          70$:    TYPE    ,69$                    ;INDICATE NO FP11-E
003312  000450                          BR      NEWPAS                  ;GO FOR SUBPASS INIT

003314  005015  020052  050106  68$:    .ASCIZ  <15><12>"* FP11-E HFP UNIT PRESENT *"<15><12>
003322  030461  042455  044040
003330  050106  052440  044516
003336  020124  051120  051505
003344  047105  020124  006452
003352  000012
003354  005015  020052  047516  69$:    .ASCIZ  <15><12>"* NO FP11-E HFP UNIT - ALL TESTS WFP ONLY *"<15><12>
003362  043040  030520  026461
003370  020105  043110  020120
003376  047125  052111  026440
003404  040440  046114  052040
003412  051505  051524  053440
003420  050106  047440  046116
003426  020131  006452  000012

                                        .EVEN

                                        ;////////////////////////////////////////////////////////////////////


                                        ;;******************************************************************
                                                ;NEW PASS ENTERS HERE
                                        ;;******************************************************************

003434  012706  001100          NEWPAS: MOV     #STACK,SP               ;RESET STACK PTR

003440  032777  010000  175476          BIT     #BIT12,@SWR             ;INHIBIT STATUS TYPEOUTS ?
003446  001011                          BNE     SUBPAS                  ;BR IF YES

003450  104401  002475                  TYPE    ,NWPAS1                 ;"PASS #"
003454  013746  001332                  MOV     $PASS,-(SP)             ;PASS COUNT INTO ...
003460  005216                          INC     (SP)                    ;  1-N RANGE
003462  104403                          TYPOS                           ;TYPE OCTAL
003464     006     000                  .BYTE   6,0                     ;  6 DIGITS, NO LEADING ZEROS
003466  104401  001321                  TYPE    ,$CRLF                  ;END THE LINE


                                        ;;******************************************************************
                                                ;NEW SUBPASS ENTERS HERE
                                        ;;******************************************************************

003472  012706  001100          SUBPAS: MOV     #STACK,SP               ;RESET SP FOR INSURANCE

003476  076600  000022                  MED     ,RWHAMI                 ;GET WHAMI INTO R0
```

```
                                                BIT     #BIT04,R0       ;1=HFP PRESENT, 0=NO
003506 001430                                   BEQ     20$             ;IF NO HFP, TEST WARM ONLY

003510 076600 000144                            MFD     ,RFLAG          ;GET FLAGS INTO R0

003514 032777 000002 175422                     BIT     #SW01,@SWR      ;SW01:  1=HFP OR WFP TEST ONLY
003522 001413                                   BEQ     1$              ;       0=ALTERNATE HFP/WFP PER PASS

003524 032777 000001 175412                     BIT     #SW00,@SWR      ;SW00:  1=WFP ONLY
003532 001403                                   BEQ     2$              ;       0=HFP ONLY
003534 042700 010000                            BIC     #BIT12,R0       ;CLEAR HFP ENABLE FLAG<5> FOR WFP
003540 000402                                   BR      3$              ;
003542 052700 010000                    2$:     BIS     #BIT12,R0       ;SET HFP ENABLE FLAG<5> FOR HFP
003546 076600 000344                    3$:     MFD     ,WFLAG          ;REWRITE FLAGS

003552 032700 010000                    1$:     BIT     #BIT12,R0       ;TEST WHO'S ENABLED: HOT, WARM
003556 001404                                   BEQ     20$             ;SET APPROPRIATE HEADER:

003560 012737 026034 024406             19$:    MOV     #ASCHOT,HOTWRM  ;"HOT: "
003566 000403                                   BR      21$             ;
003570 012737 026042 024406             20$:    MOV     #ASCWRM,HOTWRM  ;"WARM: "
003576                                  21$:
```

```
                                        ;;*****************************************************************
                                        ;*TEST 1         TEST OF FPS REGISTER BY RIPPLING A 1
                                        ;;*****************************************************************
003576  000004                  TST1:   SCOPE
003600  012700  000001                  MOV     #BIT00,R0       ; INITIAL PATTERN
003604  012737  003612  001112          MOV     #1$,$LPERR      ; ERROR LOOPING RETURN
                                
003612  010037  001170          1$:     MOV     R0,$REG0        ; EDIT PATTERN SO BIT 4 OF FPS ( FMM )
003616  042737  000020  001170          BIC     #BIT04,$REG0    ;   WILL ALWAYS BE OFF DURING TEST
003624  170137  001170                  LDFPS   $REG0           ; LOAD FPS
003630  042737  030000  001170          BIC     #BIT13!BIT12,$REG0      ; BITS 13 AND 12 ALWAYS READ 0
003636  170237  001172                  STFPS   $REG1           ; STORE FPS
003642  023737  001170  001172          CMP     $REG0,$REG1     ; LOAD/STORE WORK ?
003650  001401                          BEQ     2$              ;   YES
003652  104001                          ERROR   1               ;   NO - ERROR RETURN
003654  000241                  2$:     CLC                     ; ROTATE IN A 0
003656  006100                          ROL     R0              ; SHIFT PATTERN 1 POSITION LEFT
003660  103354                          BCC     1$              ; CONT IF NOT YET FINISHED
                                
                                
                                
                                
                                        ;;*****************************************************************
                                        ;*TEST 2         TEST OF FPS REGISTER BY RIPPLING A 0
                                        ;;*****************************************************************
003662  000004                  TST2:   SCOPE
003664  012700  177776                  MOV     #^CBIT00,R0     ; INITIAL PATTERN
003670  012737  003676  001112          MOV     #1$,$LPERR      ; ERROR LOOPING RETURN
                                
003676  010037  001170          1$:     MOV     R0,$REG0        ; EDIT PATTERN SO BIT 4 OF FPS ( FMM )
003702  042737  000020  001170          BIC     #BIT04,$REG0    ;   WILL ALWAYS BE OFF DURING TEST
003710  170137  001170                  LDFPS   $REG0           ; LOAD FPS
003714  042737  030000  001170          BIC     #BIT13!BIT12,$REG0      ; BITS 13 AND 12 ALWAYS READ 0
003722  170237  001172                  STFPS   $REG1           ; STORE FPS
003726  023737  001170  001172          CMP     $REG0,$REG1     ; LOAD/STORE WORK ?
003734  001401                          BEQ     2$              ;   YES
003736  104002                          ERROR   2               ;   NO - ERROR RETURN
003740  000261                  2$:     SEC                     ; ROTATE IN A 1
003742  006100                          ROL     R0              ; SHIFT PATTERN 1 POSITION LEFT
003744  103754                          BCS     1$              ; CONT IF NOT YET FINISHED
```

```
                                ;;*****************************************************************
                                ;*TEST 3         TEST OF SETF INSTRUCTION
                                ;;*****************************************************************
003746  000004                  TST3:   SCOPE
003750  170127  147757                  LDFPS   #FPS1           ; INITIAL PATTERN
003754  170001                          SETF                    ; TEST IT
003756  170237  001170                  STFPS   $REG0           ; GET NEW FPS
003762  022737  147557  001170          CMP     #FPS1&^CBIT07,$REG0     ; DID SETF CLEAR FPS BIT 7 ?
003770  001401                          BEQ     1$              ;
003772  104003                          ERROR   3               ;   NO - ERROR
003774                          1$:                             ;   YES - NEXT TEST
                                ;;*****************************************************************
                                ;*TEST 4         TEST OF SETD INSTRUCTION
                                ;;*****************************************************************
003774  000004                  TST4:   SCOPE
003776  170127  000000                  LDFPS   #FPS0           ; INITIAL PATTERN
004002  170011                          SETD                    ; TEST IT
004004  170237  001170                  STFPS   $REG0           ; GET NEW FPS
004010  022737  000200  001170          CMP     #FPS0!BIT07,$REG0       ; DID SETD SET FPS BIT 7 ?
004016  001401                          BEQ     1$              ;
004020  104004                          ERROR   4               ;   NO - ERROR
004022                          1$:                             ;   YES - NEXT TEST
                                ;;*****************************************************************
                                ;*TEST 5         TEST OF SETI INSTRUCTION
                                ;;*****************************************************************
004022  000004                  TST5:   SCOPE
004024  170127  147757                  LDFPS   #FPS1           ; INITIAL PATTERN
004030  170002                          SETI                    ; TEST IT
004032  170237  001170                  STFPS   $REG0           ; GET NEW FPS
004036  022737  147657  001170          CMP     #FPS1&^CBIT06,$REG0     ; DID SETI CLEAR FPS BIT 6 ?
004044  001401                          BEQ     1$              ;
004046  104005                          ERROR   5               ;   NO - ERROR
004050                          1$:                             ;   YES - NEXT TEST
                                ;;*****************************************************************
                                ;*TEST 6         TEST OF SETL INSTRUCTION
                                ;;*****************************************************************
004050  000004                  TST6:   SCOPE
004052  170127  000000                  LDFPS   #FPS0           ; INITIAL PATTERN
004056  170012                          SETL                    ; TEST IT
004060  170237  001170                  STFPS   $REG0           ; GET NEW FPS
004064  022737  000100  001170          CMP     #FPS0!BIT06,$REG0       ; DID SETL SET FPS BIT 6 ?
```

```
   731  004072  001401                    BEQ     1$              ;
   732  004074  104006                    ERROR   6               ;   NO - ERROR
   733  004076                        1$:                         ;   YES - NEXT TEST
   734
```

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

                                ;;*****************************************************************
                                ;*TEST 7         TEST OF LOAD-SRC MODE-0, STORE-DST MODE-7(PC)
                                ;;*****************************************************************
004076  000004                  TST7:   SCOPE
004100  016767  000036  175062          MOV     ADDR1,$REG0     ; GET TEST PATTERN

004106  016705  175056          MOV     $REG0,R5        ; DATA
004112  170105                  LDFPS   R5              ; M0-R5

004114  012767  001172  175052  MOV     #$REG1,$REG2    ; ADDR(DEST)
004122  170277  175046          STFPS   @$REG2          ; M7-R7

004126  026767  175036  175036  CMP     $REG0,$REG1     ; LOAD/STORE WORK?
004134  001401                  BEQ     64$             ;
004136  104007                  ERROR   7               ; NOT EQUAL, SIGNAL ERROR
004140                  64$:

004140  000401                  BR      TST10           ;;

004142  105252          ADDR1:  .WORD   105252          ; TEST PATTERN


                                ;;*****************************************************************
                                ;*TEST 10        TEST OF LOAD-SRC MODE-1, STORE-DST MODE-3
                                ;;*****************************************************************
004144  000004                  TST10:  SCOPE
004146  016767  000050  175014  MOV     ADDR2,$REG0     ; GET TEST PATTERN

004154  012703  001170          MOV     #$REG0,R3       ; ADDR(DATA)
004160  170113                  LDFPS   (R3)            ; M1-R3

004162  012767  001172  175004  MOV     #$REG1,$REG2    ; ADDR(DEST)
004170  012705  001174          MOV     #$REG2,R5       ; ADDR(ADDR(DEST))
004174  170235                  STFPS   @(R5)+          ; M3-R5

004176  020527  001176          CMP     R5,#$REG2+2     ; WAS DST ADDR REG INCRE RIGHT AMOUNT?
004202  001401                  BEQ     64$             ;
004204  104010                  ERROR   10              ; NOT EQUAL, SIGNAL ERROR
004206                  64$:

004206  026767  174756  174756  CMP     $REG0,$REG1     ; LOAD/STORE WORK?
004214  001401                  BEQ     65$             ;
004216  104011                  ERROR   11              ; NOT EQUAL, SIGNAL ERROR
004220                  65$:

004220  000401                  BR      TST11           ;;

004222  042505          ADDR2:  .WORD   042505          ; TEST PATTERN


                                ;;*****************************************************************
```

```
                                        ;*TEST 11       TEST OF LOAD-SRC MODE-2, STORE-DST MODE-5
                                        ;;*****************************************************************
004224  000004                  TST11:  SCOPE
004226  016767  000060  174734          MOV     ADDR3,$REG0     ; GET TEST PATTERN

004234  012701  001170                  MOV     #$REG0,R1       ; ADDR(DATA)
004240  170121                          LDFPS   (R1)+           ; M2-R1

004242  012767  001172  174724          MOV     #$REG1,$REG2    ; ADDR(DEST)
004250  012703  001176                  MOV     #$REG2+2,R3     ; ADDR(ADDR(DEST)+2)
004254  170253                          STFPS   @-(R3)          ; M5-R3

004256  020127  001172                  CMP     R1,#$REG0+2     ; WAS SRC ADDR REG INCRE RIGHT AMOUNT?
004262  001401                          BEQ     64$             ;
004264  104012                          ERROR   12              ; NOT EQUAL, SIGNAL ERROR
004266                          64$:

004266  020327  001174                  CMP     R3,#$REG2       ; WAS DST ADDR REG DECRE RIGHT AMOUNT?
004272  001401                          BEQ     65$             ;
004274  104013                          ERROR   13              ; NOT EQUAL, SIGNAL ERROR
004276                          65$:

004276  026767  174666  174666          CMP     $REG0,$REG1     ; LOAD/STORE WORK?
004304  001401                          BEQ     66$             ;
004306  104014                          ERROR   14              ; NOT EQUAL, SIGNAL ERROR
004310                          66$:

004310  000401                          BR      TST12           ;;

004312  105252                  ADDR3:  .WORD   105252          ; TEST PATTERN


                                        ;;*****************************************************************
                                        ;*TEST 12       TEST OF LOAD-SRC MODE-3, STORE-DST MODE-6(PC)
                                        ;;*****************************************************************
004314  000004                  TST12:  SCOPE
004316  016767  000046  174644          MOV     ADDR4,$REG0     ; GET TEST PATTERN

004324  012767  001170  174642          MOV     #$REG0,$REG2    ; ADDR(DATA)
004332  012700  001174                  MOV     #$REG2,R0       ; ADDR(ADDR(DATA))
004336  170130                          LDFPS   @(R0)+          ; M3-R0

004340  170267  174626                  STFPS   $REG1           ; M6-R7

004344  020027  001176                  CMP     R0,#$REG2+2     ; WAS SRC ADDR REG INCRE RIGHT AMOUNT?
004350  001401                          BEQ     64$             ;
004352  104015                          ERROR   15              ; NOT EQUAL, SIGNAL ERROR
004354                          64$:

004354  026767  174610  174610          CMP     $REG0,$REG1     ; LOAD/STORE WORK?
004362  001401                          BEQ     65$             ;
004364  104016                          ERROR   16              ; NOT EQUAL, SIGNAL ERROR
004366                          65$:

004366  000401                          BR      TST13           ;;
```

```
                                ADDR4:  .WORD   042505          ; TEST PATTERN

                                ;;*****************************************************************
                                ;*TEST 13        TEST OF LOAD-SRC MODE-4, STORE-DST MODE-0
                                ;;*****************************************************************
004372  000004                  TST13:  SCOPE
004374  016767  000042  174566          MOV     ADDR5,$REG0     ; GET TEST PATTERN

004402  012702  001172                  MOV     #$REG0+2,R2     ; ADDR(DATA+2)
004406  170142                          LDFPS   -(R2)           ; M4-R2

004410  170200                          STFPS   R0              ; M0-R0
004412  010067  174554                  MOV     R0,$REG1        ; DEST

004416  020227  001170                  CMP     R2,#$REG0       ; WAS SRC ADDR REG DECRE RIGHT AMOUNT?
004422  001401                          BEQ     64$             ;
004424  104017                          ERROR   17              ; NOT EQUAL, SIGNAL ERROR
004426                          64$:

004426  026767  174536  174536          CMP     $REG0,$REG1     ; LOAD/STORE WORK?
004434  001401                          BEQ     65$             ;
004436  104020                          ERROR   20              ; NOT EQUAL, SIGNAL ERROR
004440                          65$:

004440  000401                          BR      TST14           ;;

004442  105252                  ADDR5:  .WORD   105252          ; TEST PATTERN


                                ;;*****************************************************************
                                ;*TEST 14        TEST OF LOAD-SRC MODE-5, STORE-DST MODE-3(PC)
                                ;;*****************************************************************
004444  000004                  TST14:  SCOPE
004446  016767  000046  174514          MOV     ADDR6,$REG0     ; GET TEST PATTERN

004454  012767  001170  174512          MOV     #$REG0,$REG2    ; ADDR(DATA)
004462  012704  001176                  MOV     #$REG2+2,R4     ; ADDR(ADDR(DATA)+2)
004466  170154                          LDFPS   @-(R4)          ; M5-R4

004470  170237  001172                  STFPS   @#$REG1         ; M3-R7

004474  020427  001174                  CMP     R4,#$REG2       ; WAS SRC ADDR REG DECRE RIGHT AMOUNT?
004500  001401                          BEQ     64$             ;
004502  104021                          ERROR   21              ; NOT EQUAL, SIGNAL ERROR
004504                          64$:

004504  026767  174460  174460          CMP     $REG0,$REG1     ; LOAD/STORE WORK?
004512  001401                          BEQ     65$             ;
004514  104022                          ERROR   22              ; NOT EQUAL, SIGNAL ERROR
004516                          65$:

004516  000401                          BR      TST15           ;;

004520  042505                  ADDR6:  .WORD   042505          ; TEST PATTERN
```

```
                                ;;*****************************************************************
                                ;*TEST 15       TEST OF LOAD-SRC MODE-6, STORE-DST MODE-7
                                ;;*****************************************************************
004522  000004                  TST15:  SCOPE
004524  016767  000044  174436          MOV     ADDR7,$REG0     ; GET TEST PATTERN

004532  012703  001254          MOV     #$REG0+64,R3    ; ADDR(DATA)+64
004536  170163  177714          LDFPS   -64(R3)         ; M6-R3

004542  012767  001172  174424  MOV     #$REG1,$REG2    ; ADDR(DEST)
004550  012701  001134          MOV     #$REG2-40,R1    ; ADDR(ADDR(DEST))-40
004554  170271  000040          STFPS   @40(R1)         ; M7-R1

004560  026767  174404  174404  CMP     $REG0,$REG1     ; LOAD/STORE WORK?
004566  001401                  BEQ     64$             ;
004570  104014                  ERROR   14              ; NOT EQUAL, SIGNAL ERROR
004572                  64$:

004572  000401                  BR      TST16           ;;

004574  105252          ADDR7:  .WORD   105252          ; TEST PATTERN


                                ;;*****************************************************************
                                ;*TEST 16       TEST OF LOAD-SRC MODE-7, STORE-DST MODE-2(PC)
                                ;;*****************************************************************
004576  000004                  TST16:  SCOPE
004600  016767  000064  174362          MOV     ADDR10,$REG0    ; GET TEST PATTERN

004606  012767  001170  174360  MOV     #$REG0,$REG2    ; ADDR(DATA)
004614  012702  001216          MOV     #$REG2+22,R2    ; ADDR(ADDR(DATA))+22
004620  170172  177756          LDFPS   @-22(R2)        ; M7-R2

004624  012767  104117  000002  MOV     #ERROR117,1$    ; SETUP ERROR CALL FOR WRONG INCREMENT
004632  170227                  STFPS   (PC)+           ; M2-R7
004634  000000          1$:     .WORD   0               ; DEST
004636  000403                  BR      2$              ; THIS SHOULD BE NEXT INSTR AFTER FPP PC MODE 2
004640  104120                  ERROR   120             ; NOT HERE  (+4 INCRE)
004642  104121                  ERROR   121             ; OR HERE  (+6 INCRE)
004644  104122                  ERROR   122             ; OR HERE  (+10 INCRE)
004646  016767  177762  174316  2$:     MOV     1$,$REG1        ; GET DEST

004654  026767  174310  174310  CMP     $REG0,$REG1     ; LOAD/STORE WORK?
004662  001401                  BEQ     64$             ;
004664  104023                  ERROR   23              ; NOT EQUAL, SIGNAL ERROR
004666                  64$:

004666  000401                  BR      TST17           ;;

004670  042505          ADDR10: .WORD   042505          ; TEST PATTERN


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

```
  959  004672  000004                TST17:  SCOPE
  960  004674  016767  000056  174266          MOV     ADDR11,$REG0    ; GET TEST PATTERN
  961
  962  004702  016767  174262  000002          MOV     $REG0,1$        ; PUT DATA
  963  004710  170127                          LDFPS   (PC)+           ; M2-R7
  964  004712  000000                1$:       .WORD   0               ; DATA
  965  004714  000403                          BR      2$              ; THIS SHOULD BE NEXT INSTR AFTER FPP PC MODE 2
  966  004716  104120                          ERROR   120             ; NOT HERE  (+4 INCRE)
  967  004720  104121                          ERROR   121             ; OR HERE  (+6 INCRE)
  968  004722  104122                          ERROR   122             ; OR HERE  (+10 INCRE)
  969
  970  004724  012703  001174          2$:     MOV     #$REG1+2,R3     ; ADDR(DEST+2)
  971  004730  170243                          STFPS   -(R3)           ; M4-R3
  972
  973  004732  020327  001172                  CMP     R3,#$REG1       ; WAS DST ADDR REG DECRE RIGHT AMOUNT?
  974  004736  001401                          BEQ     64$             ;
  975  004740  104024                          ERROR   24              ; NOT EQUAL, SIGNAL ERROR
  976  004742                        64$:
  977
  978  004742  026767  174222  174222          CMP     $REG0,$REG1     ; LOAD/STORE WORK?
  979  004750  001401                          BEQ     65$             ;
  980  004752  104025                          ERROR   25              ; NOT EQUAL, SIGNAL ERROR
  981  004754                        65$:
  982
  983  004754  000401                          BR      TST20           ;;
  984
  985  004756  104117                ADDR11:  .WORD   ERROR117                ; TEST PATTERN
  986
  987
  988                                          ;;*****************************************************************
  989                                          ;*TEST 20       TEST OF LOAD-SRC MODE-3(PC), STORE-DST MODE-6
  990                                          ;;*****************************************************************
  991  004760  000004                TST20:  SCOPE
  992  004762  016767  000032  174200          MOV     ADDR12,$REG0    ; GET TEST PATTERN
  993
  994  004770  170137  001170                  LDFPS   @#$REG0         ; M3-R7
  995
  996  004774  012702  001144                  MOV     #$REG1-26,R2    ; ADDR(DEST)-26
  997  005000  170262  000026                  STFPS   26(R2)          ; M6-R2
  998
  999  005004  026767  174160  174160          CMP     $REG0,$REG1     ; LOAD/STORE WORK?
 1000  005012  001401                          BEQ     64$             ;
 1001  005014  104026                          ERROR   26              ; NOT EQUAL, SIGNAL ERROR
 1002  005016                        64$:
 1003
 1004  005016  000401                          BR      TST21           ;;
 1005
 1006  005020  042505                ADDR12:  .WORD   042505          ; TEST PATTERN
 1007
 1008
 1009                                          ;;*****************************************************************
 1010                                          ;*TEST 21       TEST OF LOAD-SRC MODE-6(PC), STORE-DST MODE-1
 1011                                          ;;*****************************************************************
 1012  005022  000004                TST21:  SCOPE
 1013  005024  016767  000030  174136          MOV     ADDR13,$REG0    ; GET TEST PATTERN
 1014
```

```
                        T21     TEST OF LOAD-SRC MODE-6(PC), STORE-DST MODE-1

1015  005032  170167  174132            LDFPS   $REG0           ; M6-R7
1016
1017  005036  012702  001172            MOV     #$REG1,R2       ; ADDR(DEST)
1018  005042  170212                    STFPS   (R2)            ; M1-R2
1019
1020  005044  026767  174120  174120    CMP     $REG0,$REG1     ; LOAD/STORE WORK?
1021  005052  001401                    BEQ     64$             ;
1022  005054  104026                    ERROR   26              ; NOT EQUAL, SIGNAL ERROR
1023  005056                    64$:
1024
1025  005056  000401                    BR      TST22           ;;
1026
1027  005060  105252            ADDR13: .WORD   105252          ; TEST PATTERN
1028
1029
1030                            ;;**************************************************************
1031                            ;*TEST 22        TEST OF LOAD-SRC MODE-7(PC), STORE-DST MODE-2
1032                            ;;**************************************************************
1033  005062  000004            TST22:  SCOPE
1034  005064  016767  000046  174076    MOV     ADDR14,$REG0    ; GET TEST PATTERN
1035
1036  005072  012767  001170  174074    MOV     #$REG0,$REG2    ; ADDR(DATA)
1037  005100  170177  174070            LDFPS   @$REG2          ; M7-R7
1038
1039  005104  012704  001172            MOV     #$REG1,R4       ; ADDR(DEST)
1040  005110  170224                    STFPS   (R4)+           ; M2-R4
1041
1042  005112  020427  001174            CMP     R4,#$REG1+2     ; WAS DST ADDR REG INCRE RIGHT AMOUNT?
1043  005116  001401                    BEQ     64$             ;
1044  005120  104027                    ERROR   27              ; NOT EQUAL, SIGNAL ERROR
1045  005122                    64$:
1046
1047  005122  026767  174042  17 042    CMP     $REG0,$REG1     ; LOAD/STORE WORK?
1048  005130  001401                    BEQ     65$             ;
1049  005132  104022                    ERROR   22              ; NOT EQUAL, SIGNAL ERROR
1050  005134                    65$:
1051
1052  005134  000401                    BR      TST23           ;;
1053
1054  005136  042505            ADDR14: .WORD   042505          ; TEST PATTERN
1055
1056
```

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

                                        ;;*****************************************************************
                                        ;*TEST 23       TEST OF CFCC INSTRUCTION
                                        ;;*****************************************************************
005140  000004                  TST23:  SCOPE
005142  012700  000010                  MOV     #10,R0          ; NUMBER OF ENTRIES IN TABLE
005146  012701  005252                  MOV     #OFCC,R1        ; START OF TABLE
005152  012737  005160  001112          MOV     #1$,$LPERR      ; ERROR LOOPING RETURN

005160  011137  001170          1$:     MOV     (R1),$REG0      ; GET FIRST FCC PATTERN
005164  042737  177760  001170          BIC     #CCONLY,$REG0   ;  BITS IN FCC POSITIONS ONLY
005172  170137  001170                  LDFPS   $REG0           ; STORE IN FPS REGISTER
005176  032711  000040                  BIT     #BIT05,(R1)     ; TEST WHETHER FLOATING A 1 OR 0
005202  001002                          BNE     2$              ;
005204  000257                          CCC                     ; FOR FLOAT A 1, START W/ CC = 0000
005206  000401                          BR      3$              ;
005210  000277                  2$:     SCC                     ; FOR FLOAT A 0, START W/ CC = 1111
005212  170000                  3$:     CFCC                    ; COPY THE CONDITION CODES
005214  013737  177776  001172          MOV     @#PSW,$REG1     ; GET CPU CC BITS
005222  042737  177760  001172          BIC     #CCONLY,$REG1   ; CLEAR EXTRANEOUS BITS
005230  023737  001170  001172          CMP     $REG0,$REG1     ; WERE THEY COPIED OK ?
005236  001401                          BEQ     4$              ;
005240  104030                          ERROR   30              ; NO - SIGNAL ERROR
005242  062701  000002          4$:     ADD     #2,R1           ; INCRE R1 OUT OF ERROR LOOP
005246  077034                          SOB     R0,1$           ; LOOP CONTROL
005250  000410                          BR      TST24           ;;

005252  000001  000002  000004  OFCC:   .WORD   000001,000002,000004,000010     ; TABLE OF CC
005260  000010
005262  000056  000055  000053          .WORD   000056,000055,000053,000047     ;   TEST PATTERNS
005270  000047
```

```
1092                                    ;;*****************************************************************
1093                                    ;*TEST 24       TEST FP ACO, D MODE, TEST PATTERN LDAD-1
1094                                    ;;*****************************************************************
1095  005272  000004                    TST24:  SCOPE
1096  005274  012704  021214                    MOV     #LDARDO,R4      ; PTR TO TESTING ROUTINE
1097  005300  012705  005312                    MOV     #LDAD1,R5       ; PTR TO TEST DATA SET
1098  005304  004737  021032                    JSR     PC,@#LDADT      ; GO TEST
1099
1100  005310  000413                          BR      TST25           ;;
1101
1102  005312                          LDAD1:  ; TEST DATA SET LDAD-1:
1103  005312  000000  000000  000000            .WORD   0,   0,   0,   0        ; PATTERN 1
1104  005320  000000
1105  005322  000000  000000  000000            .WORD   0,   0,   0,   0        ; PATTERN EXPECTED AFTER
1106  005330  000000
1107  005332  047653  047644                    .WORD   047653,047644   ; FPS:  BEFORE, AFTER
1108  005336  000000                            .WORD   000000          ; FEC AFTER ( 0 = N/A )
1109
1110                                    ;;*****************************************************************
1111                                    ;*TEST 25       TEST FP ACO, D MODE, TEST PATTERN LDAD-2
1112                                    ;;*****************************************************************
1113  005340  000004                    TST25:  SCOPE
1114  005342  012704  021214                    MOV     #LDARDO,R4      ; PTR TO TESTING ROUTINE
1115  005346  012705  005360                    MOV     #LDAD2,R5       ; PTR TO TEST DATA SET
1116  005352  004737  021032                    JSR     PC,@#LDADT      ; GO TEST
1117
1118  005356  000413                          BR      TST26           ;;
1119
1120  005360                          LDAD2:  ; TEST DATA SET LDAD-2:
1121  005360  177777  177777  177777            .WORD   M1,  M1,  M1,  M1       ; PATTERN 2
1122  005366  177777
1123  005370  177777  177777  177777            .WORD   M1,  M1,  M1,  M1       ; PATTERN EXPECTED AFTER
1124  005376  177777
1125  005400  047745  047750                    .WORD   047745,047750   ; FPS:  BEFORE, AFTER
1126  005404  000000                            .WORD   000000          ; FEC AFTER ( 0 = N/A )
1127
1128                                    ;;*****************************************************************
1129                                    ;*TEST 26       TEST FP ACO, D MODE, TEST PATTERN LDAD-3
1130                                    ;;*****************************************************************
1131  005406  000004                    TST26:  SCOPE
1132  005410  012704  021214                    MOV     #LDARDO,R4      ; PTR TO TESTING ROUTINE
1133  005414  012705  005426                    MOV     #LDAD3,R5       ; PTR TO TEST DATA SET
1134  005420  004737  021032                    JSR     PC,@#LDADT      ; GO TEST
1135
1136  005424  000413                          BR      TST27           ;;
1137
1138  005426                          LDAD3:  ; TEST DATA SET LDAD-3:
1139  005426  052525  052525  052525            .WORD   ALTP,ALTP,ALTP,ALTP     ; PATTERN 3
1140  005434  052525
1141  005436  052525  052525  052525            .WORD   ALTP,ALTP,ALTP,ALTP     ; PATTERN EXPECTED AFTER
1142  005444  052525
1143  005446  047652  047640                    .WORD   047652,047640   ; FPS:  BEFORE, AFTER
1144  005452  000000                            .WORD   000000          ; FEC AFTER ( 0 = N/A )
1145
1146                                    ;;*****************************************************************
1147                                    ;*TEST 27       TEST FP ACO, D MODE, TEST PATTERN LDAD-4
```

```
1148
1149  005454  000004                  ;;*****************************************************************
1150  005456  012704  021214          TST27:  SCOPE
1151  005462  012705  005474                  MOV     #LDARD0,R4      ; PTR TO TESTING ROUTINE
1152  005466  004737  021032                  MOV     #LDAD4,R5       ; PTR TO TEST DATA SET
1153                                          JSR     PC,@#LDADT      ; GO TEST
1154  005472  000413                         BR      TST30           ;;
1155
1156  005474                          LDAD4:  ; TEST DATA SET LDAD-4:
1157  005474  125252  125252  125252          .WORD   ALTN,ALTN,ALTN,ALTN     ; PATTERN 4
1158  005502  125252
1159  005504  125252  125252  125252          .WORD   ALTN,ALTN,ALTN,ALTN     ; PATTERN EXPECTED AFTER
1160  005512  125252
1161  005514  047704  047710                 .WORD   047704,047710   ; FPS:  BEFORE, AFTER
1162  005520  000000                         .WORD   000000          ; FEC AFTER ( 0 = N/A )
1163
1164                                  ;;*****************************************************************
1165                                  ;*TEST 30       TEST FP AC0, D MODE, TEST PATTERN LDAD-5
1166                                  ;;*****************************************************************
1167  005522  000004                  TST30:  SCOPE
1168  005524  012704  021214                  MOV     #LDARD0,R4      ; PTR TO TESTING ROUTINE
1169  005530  012705  005542                  MOV     #LDAD5,R5       ; PTR TO TEST DATA SET
1170  005534  004737  021032                  JSR     PC,@#LDADT      ; GO TEST
1171
1172  005540  000413                         BR      TST31           ;;
1173
1174  005542                          LDAD5:  ; TEST DATA SET LDAD-5:
1175  005542  100177  177777  177777          .WORD   ZX1MN,M1,M1,M1          ; PATTERN 5
1176  005550  177777
1177  005552  100177  177777  177777          .WORD   ZX1MN,M1,M1,M1          ; PATTERN EXPECTED AFTER
1178  005560  177777
1179  005562  043743  043754                 .WORD   043743,043754   ; FPS:  BEFORE, AFTER
1180  005566  000000                         .WORD   000000          ; FEC AFTER ( 0 = N/A )
1181
1182                                  ;;*****************************************************************
1183                                  ;*TEST 31       TEST FP AC0, D MODE, TEST PATTERN LDAD-6
1184                                  ;;*****************************************************************
1185  005570  000004                  TST31:  SCOPE
1186  005572  012704  021214                  MOV     #LDARD0,R4      ; PTR TO TESTING ROUTINE
1187  005576  012705  005610                  MOV     #LDAD6,R5       ; PTR TO TEST DATA SET
1188  005602  004737  021032                  JSR     PC,@#LDADT      ; GO TEST
1189
1190  005606  000413                         BR      TST32           ;;
1191
1192  005610                          LDAD6:  ; TEST DATA SET LDAD-6:
1193  005610  100000  000000  000000          .WORD   M0,  0,   0,   0        ; PATTERN 6
1194  005616  000000
1195  005620  052525  177777  125252          .WORD   ALTP,M1,  ALTN,0        ; PATTERN EXPECTED AFTER
1196  005626  000000
1197  005630  047602  147614                 .WORD   047602,147614   ; FPS:  BEFORE, AFTER
1198  005634  100014                         .WORD   100014          ; FEC AFTER ( 0 = N/A )
1199
1200                                  ;;*****************************************************************
1201                                  ;*TEST 32       TEST FP AC1, D MODE, TEST PATTERN LDAD-1
1202                                  ;;*****************************************************************
1203  005636  000004                  TST32:  SCOPE
```

```
1204  005640  012704  021230          T32     TEST FP AC1, D MODE, TEST PATTERN LDAD-1
1205  005644  012705  005312                  MOV     #LDARD1,R4      ; PTR TO TESTING ROUTINE
1206  005650  004737  021032                  MOV     #LDAD1,R5       ; PTR TO TEST DATA SET
1207                                          JSR     PC,@#LDADT      ; GO TEST
1208  005654  000400                         BR      TST33           ;;
1209
1210                                  ; ( FOR DATA SEE TEST OF FP REG AC0 WITH TEST PATTERN LDAD-1, ABOVE )
1211
1212                                  ;;*********************************************************************
1213                                  ;*TEST 33        TEST FP AC1, D MODE, TEST PATTERN LDAD-2
1214                                  ;;*********************************************************************
1215  005656  000004                 TST33:  SCOPE
1216  005660  012704  021230                  MOV     #LDARD1,R4      ; PTR TO TESTING ROUTINE
1217  005664  012705  005360                  MOV     #LDAD2,R5       ; PTR TO TEST DATA SET
1218  005670  004737  021032                  JSR     PC,@#LDADT      ; GO TEST
1219
1220  005674  000400                         BR      TST34           ;;
1221
1222                                  ; ( FOR DATA SEE TEST OF FP REG AC0 WITH TEST PATTERN LDAD-2, ABOVE )
1223
1224                                  ;;*********************************************************************
1225                                  ;*TEST 34        TEST FP AC1, D MODE, TEST PATTERN LDAD-3
1226                                  ;;*********************************************************************
1227  005676  000004                 TST34:  SCOPE
1228  005700  012704  021230                  MOV     #LDARD1,R4      ; PTR TO TESTING ROUTINE
1229  005704  012705  005426                  MOV     #LDAD3,R5       ; PTR TO TEST DATA SET
1230  005710  004737  021032                  JSR     PC,@#LDADT      ; GO TEST
1231
1232  005714  000400                         BR      TST35           ;;
1233
 234                                  ; ( FOR DATA SEE TEST OF FP REG AC0 WITH TEST PATTERN LDAD-3, ABOVE )
1235
1236                                  ;;*********************************************************************
1237                                  ;*TEST 35        TEST FP AC1, D MODE, TEST PATTERN LDAD-4
1238                                  ;;*********************************************************************
1239  005716  000004                 TST35:  SCOPE
1240  005720  012704  021230                  MOV     #LDARD1,R4      ; PTR TO TESTING ROUTINE
1241  005724  012705  005474                  MOV     #LDAD4,R5       ; PTR TO TEST DATA SET
1242  005730  004737  021032                  JSR     PC,@#LDADT      ; GO TEST
1243
1244  005734  000400                         BR      TST36           ;;
1245
1246                                  ; ( FOR DATA SEE TEST OF FP REG AC0 WITH TEST PATTERN LDAD-4, ABOVE )
1247
1248                                  ;;*********************************************************************
1249                                  ;*TEST 36        TEST FP AC1, D MODE, TEST PATTERN LDAD-5
1250                                  ;;*********************************************************************
1251  005736  000004                 TST36:  SCOPE
1252  005740  012704  021230                  MOV     #LDARD1,R4      ; PTR TO TESTING ROUTINE
1253  005744  012705  005542                  MOV     #LDAD5,R5       ; PTR TO TEST DATA SET
1254  005750  004737  021032                  JSR     PC,@#LDADT      ; GO TEST
1255
1256  005754  000400                         BR      TST37           ;;
1257
1258                                  ; ( FOR DATA SEE TEST OF FP REG AC0 WITH TEST PATTERN LDAD-5, ABOVE )
1259
```

```
                                ;;*****************************************************************
                                ;*TEST 37        TEST FP AC1, D MODE, TEST PATTERN LDAD-6
                                ;*****************************************************************
005756  000004                  TST37:  SCOPE
005760  012704  021230                  MOV     #LDARD1,R4      ; PTR TO TESTING ROUTINE
005764  012705  005610                  MOV     #LDAD6,R5       ; PTR TO TEST DATA SET
005770  004737  021032                  JSR     PC,@#LDADT      ; GO TEST

005774  000400                          BR      TST40           ;;

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


                                ;;*****************************************************************
                                ;*TEST 40        TEST FP AC2, D MODE, TEST PATTERN LDAD-1
                                ;*****************************************************************
005776  000004                  TST40:  SCOPE
006000  012704  021244                  MOV     #LDARD2,R4      ; PTR TO TESTING ROUTINE
006004  012705  005312                  MOV     #LDAD1,R5       ; PTR TO TEST DATA SET
006010  004737  021032                  JSR     PC,@#LDADT      ; GO TEST

006014  000400                          BR      TST41           ;;

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


                                ;;*****************************************************************
                                ;*TEST 41        TEST FP AC2, D MODE, TEST PATTERN LDAD-2
                                ;*****************************************************************
006016  000004                  TST41:  SCOPE
006020  012704  021244                  MOV     #LDARD2,R4      ; PTR TO TESTING ROUTINE
006024  012705  005360                  MOV     #LDAD2,R5       ; PTR TO TEST DATA SET
006030  004737  021032                  JSR     PC,@#LDADT      ; GO TEST

006034  000400                          BR      TST42           ;;

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


                                ;;*****************************************************************
                                ;*TEST 42        TEST FP AC2, D MODE, TEST PATTERN LDAD-3
                                ;*****************************************************************
006036  000004                  TST42:  SCOPE
006040  012704  021244                  MOV     #LDARD2,R4      ; PTR TO TESTING ROUTINE
006044  012705  005426                  MOV     #LDAD3,R5       ; PTR TO TEST DATA SET
006050  004737  021032                  JSR     PC,@#LDADT      ; GO TEST

006054  000400                          BR      TST43           ;;

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


                                ;;*****************************************************************
                                ;*TEST 43        TEST FP AC2, D MODE, TEST PATTERN LDAD-4
                                ;*****************************************************************
006056  000004                  TST43:  SCOPE
006060  012704  021244                  MOV     #LDARD2,R4      ; PTR TO TESTING ROUTINE
006064  012705  005474                  MOV     #LDAD4,R5       ; PTR TO TEST DATA SET
006070  004737  021032                  JSR     PC,@#LDADT      ; GO TEST
```

```
1316
1317  006074  000400                        BR      TST44           ;;
1318
1319                                  ; ( FOR DATA SEE TEST OF FP REG AC0 WITH TEST PATTERN LDAD-4, ABOVE )
1320
1321                                  ;;*****************************************************************
1322                                  ;*TEST 44       TEST FP AC2, D MODE, TEST PATTERN LDAD-5
1323                                  ;;*****************************************************************
1324  006076  000004                  TST44:  SCOPE
1325  006100  012704  021244                  MOV     #LDARD2,R4      ; PTR TO TESTING ROUTINE
1326  006104  012705  005542                  MOV     #LDAD5,R5       ; PTR TO TEST DATA SET
1327  006110  004737  021032                  JSR     PC,@#LDADT      ; GO TEST
1328
1329  006114  000400                        BR      TST45           ;;
1330
1331                                  ; ( FOR DATA SEE TEST OF FP REG AC0 WITH TEST PATTERN LDAD-5, ABOVE )
1332
1333                                  ;;*****************************************************************
1334                                  ;*TEST 45       TEST FP AC2, D MODE, TEST PATTERN LDAD-6
1335                                  ;;*****************************************************************
1336  006116  000004                  TST45:  SCOPE
1337  006120  012704  021244                  MOV     #LDARD2,R4      ; PTR TO TESTING ROUTINE
1338  006124  012705  005610                  MOV     #LDAD6,R5       ; PTR TO TEST DATA SET
1339  006130  004737  021032                  JSR     PC,@#LDADT      ; GO TEST
1340
1341  006134  000400                        BR      TST46           ;;
1342
1343                                  ; ( FOR DATA SEE TEST OF FP REG AC0 WITH TEST PATTERN LDAD-6, ABOVE )
1344
1345
1346                                  ;;*****************************************************************
1347                                  ;*TEST 46       TEST FP AC3, D MODE, TEST PATTERN LDAD-1
1348                                  ;;*****************************************************************
1349  006136  000004                  TST46:  SCOPE
1350  006140  012704  021260                  MOV     #LDARD3,R4      ; PTR TO TESTING ROUTINE
1351  006144  012705  005312                  MOV     #LDAD1,R5       ; PTR TO TEST DATA SET
1352  006150  004737  021032                  JSR     PC,@#LDADT      ; GO TEST
1353
1354  006154  000400                        BR      TST47           ;;
1355
1356                                  ; ( FOR DATA SEE TEST OF FP REG AC0 WITH TEST PATTERN LDAD-1, ABOVE )
1357
1358                                  ;;*****************************************************************
1359                                  ;*TEST 47       TEST FP AC3, D MODE, TEST PATTERN LDAD-2
1360                                  ;;*****************************************************************
1361  006156  000004                  TST47:  SCOPE
1362  006160  012704  021260                  MOV     #LDARD3,R4      ; PTR TO TESTING ROUTINE
1363  006164  012705  005360                  MOV     #LDAD2,R5       ; PTR TO TEST DATA SET
1364  006170  004737  021032                  JSR     PC,@#LDADT      ; GO TEST
1365
1366  006174  000400                        BR      TST50           ;;
1367
1368                                  ; ( FOR DATA SEE TEST OF FP REG AC0 WITH TEST PATTERN LDAD-2, ABOVE )
1369
1370                                  ;;*****************************************************************
1371                                  ;*TEST 50       TEST FP AC3, D MODE, TEST PATTERN LDAD-3
```

```
1372                                    ;;*****************************************************************
1373  006176  000004                    TST50:  SCOPE
1374  006200  012704  021260                   MOV     #LDARD3,R4      ; PTR TO TESTING ROUTINE
1375  006204  012705  005426                   MOV     #LDAD3,R5       ; PTR TO TEST DATA SET
1376  006210  004737  021032                   JSR     PC,@#LDADT      ; GO TEST
1377
1378  006214  000400                           BR      TST51           ;;
1379
1380                                    ; ( FOR DATA SEE TEST OF FP REG AC0 WITH TEST PATTERN LDAD-3, ABOVE )
1381
1382                                    ;;*****************************************************************
1383                                    ;*TEST 51        TEST FP AC3, D MODE, TEST PATTERN LDAD-4
1384                                    ;;*****************************************************************
1385  006216  000004                    TST51:  SCOPE
1386  006220  012704  021260                   MOV     #LDARD3,R4      ; PTR TO TESTING ROUTINE
1387  006224  012705  005474                   MOV     #LDAD4,R5       ; PTR TO TEST DATA SET
1388  006230  004737  021032                   JSR     PC,@#LDADT      ; GO TEST
1389
1390  006234  000400                           BR      TST52           ;;
1391
1392                                    ; ( FOR DATA SEE TEST OF FP REG AC0 WITH TEST PATTERN LDAD-4, ABOVE )
1393
1394                                    ;;*****************************************************************
1395                                    ;*TEST 52        TEST FP AC3, D MODE, TEST PATTERN LDAD-5
1396                                    ;;*****************************************************************
1397  006236  000004                    TST52:  SCOPE
1398  006240  012704  021260                   MOV     #LDARD3,R4      ; PTR TO TESTING ROUTINE
1399  006244  012705  005542                   MOV     #LDAD5,R5       ; PTR TO TEST DATA SET
1400  006250  004737  021032                   JSR     PC,@#LDADT      ; GO TEST
1401
1402  006254  000400                           BR      TST53           ;;
1403
1404                                    ; ( FOR DATA SEE TEST OF FP REG AC0 WITH TEST PATTERN LDAD-5, ABOVE )
1405
1406                                    ;;*****************************************************************
1407                                    ;*TEST 53        TEST FP AC3, D MODE, TEST PATTERN LDAD-6
1408                                    ;;*****************************************************************
1409  006256  000004                    TST53:  SCOPE
1410  006260  012704  021260                   MOV     #LDARD3,R4      ; PTR TO TESTING ROUTINE
1411  006264  012705  005610                   MOV     #LDAD6,R5       ; PTR TO TEST DATA SET
1412  006270  004737  021032                   JSR     PC,@#LDADT      ; GO TEST
1413
1414  006274  000400                           BR      TST54           ;;
1415
1416                                    ; ( FOR DATA SEE TEST OF FP REG AC0 WITH TEST PATTERN LDAD-6, ABOVE )
1417
1418
1419                                    ;;*****************************************************************
1420                                    ;*TEST 54        TEST FP AC4, D MODE, TEST PATTERN LDAD-1
1421                                    ;;*****************************************************************
1422  006276  000004                    TST54:  SCOPE
1423  006300  012704  021274                   MOV     #LDARD4,R4      ; PTR TO TESTING ROUTINE
1424  006304  012705  005312                   MOV     #LDAD1,R5       ; PTR TO TEST DATA SET
1425  006310  004737  021032                   JSR     PC,@#LDADT      ; GO TEST
1426
1427  006314  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
                                        ;;*****************************************************************
006316  000004                          TST55:  SCOPE
006320  012704  021274                          MOV     #LDAF04,R4      ; PTR TO TESTING ROUTINE
006324  012705  005360                          MOV     #LDAD2,R5       ; PTR TO TEST DATA SET
006330  004737  021032                          JSR     PC,@#LDADT      ; GO TEST

006334  000400                                  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
                                        ;;*****************************************************************
006336  000004                          TST56:  SCOPE
006340  012704  021274                          MOV     #LDARD4,R4      ; PTR TO TESTING ROUTINE
006344  012705  005426                          MOV     #LDAD3,R5       ; PTR TO TEST DATA SET
006350  004737  021032                          JSR     PC,@#LDADT      ; GO TEST

006354  000400                                  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
                                        ;;*****************************************************************
006356  000004                          TST57:  SCOPE
006360  012704  021274                          MOV     #LDARD4,R4      ; PTR TO TESTING ROUTINE
006364  012705  005474                          MOV     #LDAD4,R5       ; PTR TO TEST DATA SET
006370  004737  021032                          JSR     PC,@#LDADT      ; GO TEST

006374  000400                                  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
                                        ;;*****************************************************************
006376  000004                          TST60:  SCOPE
006400  012704  021274                          MOV     #LDARD4,R4      ; PTR TO TESTING ROUTINE
006404  012705  005542                          MOV     #LDAD5,R5       ; PTR TO TEST DATA SET
006410  004737  021032                          JSR     PC,@#LDADT      ; GO TEST

006414  000400                                  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
                                        ;;*****************************************************************
006416  000004                          TST61:  SCOPE
```

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

1484  006420  012704  021314            MOV     #LDARD5,R4      ; PTR TO TESTING ROUTINE
1485  006424  012705  005312            MOV     #LDAD1,R5       ; PTR TO TEST DATA SET
1486  006430  004737  021032            JSR     PC,@#LDADT      ; GO TEST
1487
1488  006434  000400                    BR      TST62           ;;
1489
1490                          ; ( FOR DATA SEE TEST OF FP REG AC0 WITH TEST PATTERN LDAD-1, ABOVE )
1491
1492                          ;;*****************************************************************
1493                          ;*TEST 62       TEST FP AC5, D MODE, TEST PATTERN LDAD-2
1494                          ;;*****************************************************************
1495  006436  000004          TST62:  SCOPE
1496  006440  012704  021314            MOV     #LDARD5,R4      ; PTR TO TESTING ROUTINE
1497  006444  012705  005360            MOV     #LDAD2,R5       ; PTR TO TEST DATA SET
1498  006450  004737  021032            JSR     PC,@#LDADT      ; GO TEST
1499
1500  006454  000400                    BR      TST63           ;;
1501
1502                          ; ( FOR DATA SEE TEST OF FP REG AC0 WITH TEST PATTERN LDAD-2, ABOVE )
1503
1504                          ;;*****************************************************************
1505                          ;*TEST 63       TEST FP AC5, D MODE, TEST PATTERN LDAD-3
1506                          ;;*****************************************************************
1507  006456  000004          TST63:  SCOPE
1508  006460  012704  021314            MOV     #LDARD5,R4      ; PTR TO TESTING ROUTINE
1509  006464  012705  005426            MOV     #LDAD3,R5       ; PTR TO TEST DATA SET
1510  006470  004737  021032            JSR     PC,@#LDADT      ; GO TEST
1511
1512  006474  000400                    BR      TST64           ;;
1513
1514                          ; ( FOR DATA SEE TEST OF FP REG AC0 WITH TEST PATTERN LDAD-3, ABOVE )
1515
1516                          ;;*****************************************************************
1517                          ;*TEST 64       TEST FP AC5, D MODE, TEST PATTERN LDAD-4
1518                          ;;*****************************************************************
1519  006476  000004          TST64:  SCOPE
1520  006500  012704  021314            MOV     #LDARD5,R4      ; PTR TO TESTING ROUTINE
1521  006504  012705  005474            MOV     #LDAD4,R5       ; PTR TO TEST DATA SET
1522  006510  004737  021032            JSR     PC,@#LDADT      ; GO TEST
1523
1524  006514  000400                    BR      TST65           ;;
1525
1526                          ; ( FOR DATA SEE TEST OF FP REG AC0 WITH TEST PATTERN LDAD-4, ABOVE )
1527
1528                          ;;*****************************************************************
1529                          ;*TEST 65       TEST FP AC5, D MODE, TEST PATTERN LDAD-5
1530                          ;;*****************************************************************
1531  006516  000004          TST65:  SCOPE
1532  006520  012704  021314            MOV     #LDARD5,R4      ; PTR TO TESTING ROUTINE
1533  006524  012705  005542            MOV     #LDAD5,R5       ; PTR TO TEST DATA SET
1534  006530  004737  021032            JSR     PC,@#LDADT      ; GO TEST
1535
1536  006534  000400                    BR      TST66           ;;
1537
1538                          ; ( FOR DATA SEE TEST OF FP REG AC0 WITH TEST PATTERN LDAD-5, ABOVE )
1539
```

```
1540
1541
```

```
                                        ;;*****************************************************************
                                        ;*TEST 66       TEST FP AC0, F MODE, TEST PATTERN LDAF-1
                                        ;;*****************************************************************
006536  000004                  TST66:  SCOPE
006540  012704  021514                  MOV     #LDARF0,R4      ; PTR TO TESTING ROUTINE
006544  012705  006556                  MOV     #LDAF1,R5       ; PTR TO TEST DATA SET
006550  004737  021334                  JSR     PC,@#LDAFT      ; GO TEST
006554  000415                          BR      TST67           ;;
006556                          LDAF1:  ; TEST DATA SET LDAF-1:
006556  000000  000000  000000          .WORD   0,0,0,0                 ; LDD PATTERN
006564  000000
006566  177777  177777                  .WORD   M1,M1                   ; LDF PATTERN
006572  177777  177777  000000          .WORD   M1,M1,0,0               ; STD EXPECTED PATTERN
006600  000000
006602  047547  047750                  .WORD   047547,047750   ; FPS:  BEFORE, AFTER
006606  000000                          .WORD   000000          ; FEC AFTER ( 0 = N/A )

                                        ;;*****************************************************************
                                        ;*TEST 67       TEST FP AC0, F MODE, TEST PATTERN LDAF-2
                                        ;;*****************************************************************
006610  000004                  TST67:  SCOPE
006612  012704  021514                  MOV     #LDARF0,R4      ; PTR TO TESTING ROUTINE
006616  012705  006630                  MOV     #LDAF2,R5       ; PTR TO TEST DATA SET
006622  004737  021334                  JSR     PC,@#LDAFT      ; GO TEST
006626  000415                          BR      TST70           ;;
006630                          LDAF2:  ; TEST DATA SET LDAF-2:
006630  177777  177777  177777          .WORD   M1,M1,M1,M1             ; LDD PATTERN
006636  177777
006640  000000  000000                  .WORD   0,0                     ; LDF PATTERN
006644  000000  000000  177777          .WORD   0,0,M1,M1               ; STD EXPECTED PATTERN
006652  177777
006654  047413  047604                  .WORD   047413,047604   ; FPS:  BEFORE, AFTER
006660  000000                          .WORD   000000          ; FEC AFTER ( 0 = N/A )

                                        ;;*****************************************************************
                                        ;*TEST 70       TEST FP AC0, F MODE, TEST PATTERN LDAF-3
                                        ;;*****************************************************************
006662  000004                  TST70:  SCOPE
006664  012704  021514                  MOV     #LDARF0,R4      ; PTR TO TESTING ROUTINE
006670  012705  006702                  MOV     #LDAF3,R5       ; PTR TO TEST DATA SET
006674  004737  021334                  JSR     PC,@#LDAFT      ; GO TEST
006700  000415                          BR      TST71           ;;
006702                          LDAF3:  ; TEST DATA SET LDAF-3:
006702  052525  052525  052525          .WORD   ALTP,ALTP,ALTP,ALTP     ; LDD PATTERN
006710  052525
006712  125252  125252                  .WORD   ALTN,ALTN               ; LDF PATTERN
006716  125252  125252  052525          .WORD   ALTN,ALTN,ALTP,ALTP     ; STD EXPECTED PATTERN
006724  052525
006726  047547  047750                  .WORD   047547,047750   ; FPS:  BEFORE, AFTER
006732  000000                          .WORD   000000          ; FEC AFTER ( 0 = N/A )
```

```
1598
1599                                    ;;*****************************************************************
1600                                    ;*TEST 71        TEST FP AC0, F MODE, TEST PATTERN LDAF-4
1601                                    ;;*****************************************************************
1602  006734  000004                   TST71:  SCOPE
1603  006736  012704  021514                   MOV     #LDARF0,R4      ; PTR TO TESTING ROUTINE
1604  006742  012705  006754                   MOV     #LDAF4,R5       ; PTR TO TEST DATA SET
1605  006746  004737  021334                   JSR     PC,@#LDAFT      ; GO TEST
1606
1607  006752  000415                           BR      TST72           ;;
1608
1609  006754                           LDAF4:  ; TEST DATA SET LDAF-4:
1610  006754  125252  125252  125252           .WORD   ALTN,ALTN,ALTN,ALTN     ; LDD PATTERN
1611  006762  125252
1612  006764  052525  052525                   .WORD   ALTP,ALTP               ; LDF PATTERN
1613  006770  052525  052525  125252           .WORD   ALTP,ALTP,ALTN,ALTN     ; STD EXPECTED PATTERN
1614  006776  125252
1615  007000  047417  047600                   .WORD   047417,047600   ; FPS:  BEFORE, AFTER
1616  007004  000000                           .WORD   000000          ; FEC AFTER ( 0 = N/A )
1617
1618                                    ;;*****************************************************************
1619                                    ;*TEST 72        TEST FP AC0, F MODE, TEST PATTERN LDAF-5
1620                                    ;;*****************************************************************
1621  007006  000004                   TST72:  SCOPE
1622  007010  012704  021514                   MOV     #LDARF0,R4      ; PTR TO TESTING ROUTINE
1623  007014  012705  007026                   MOV     #LDAF5,R5       ; PTR TO TEST DATA SET
1624  007020  004737  021334                   JSR     PC,@#LDAFT      ; GO TEST
1625
1626  007024  000415                           BR      TST73           ;;
1627
1628  007026                           LDAF5:  ; TEST DATA SET LDAF-5:
1629  007026  177777  177777  177777           .WORD   M1,M1,M1,M1             ; LDD PATTERN
1630  007034  177777
1631  007036  100000  000000                   .WORD   M0,00                   ; LDF PATTERN
1632  007042  100000  000000  177777           .WORD   M0,00,M1,M1             ; STD EXPECTED PATTERN
1633  007050  177777
1634  007052  043443  043654                   .WORD   043443,043654   ; FPS:  BEFORE, AFTER
1635  007056  000000                           .WORD   000000          ; FEC AFTER ( 0 = N/A )
1636
1637                                    ;;*****************************************************************
1638                                    ;*TEST 73        TEST FP AC0, F MODE, TEST PATTERN LDAF-6
1639                                    ;;*****************************************************************
1640  007060  000004                   TST73:  SCOPE
1641  007062  012704  021514                   MOV     #LDARF0,R4      ; PTR TO TESTING ROUTINE
1642  007066  012705  007100                   MOV     #LDAF6,R5       ; PTR TO TEST DATA SET
1643  007072  004737  021334                   JSR     PC,@#LDAFT      ; GO TEST
1644
1645  007076  000415                           BR      TST74           ;;
1646
1647  007100                           LDAF6:  ; TEST DATA SET LDAF-6:
1648  007100  052525  000000  125252           .WORD   ALTP,0,ALTN,M1          ; LDD PATTERN
1649  007106  177777
1650  007110  100000  000000                   .WORD   M0,0                    ; LDF PATTERN
1651  007114  052525  000000  125252           .WORD   ALTP,0,ALTN,M1          ; STD EXPECTED PATTERN
1652  007122  177777
1653  007124  047543  147754                   .WORD   047543,147754   ; FPS:  BEFORE, AFTER
```

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

1654  007130  100014                    .WORD   100014          ; FEC AFTER ( 0 = N/A )
1655
1656
1657                            ;;*****************************************************************
1658                            ;*TEST 74       TEST FP AC1, F MODE, TEST PATTERN LDAF-1
1659                            ;*****************************************************************
1660  007132  000004            TST74:  SCOPE
1661  007134  012704  021536            MOV     #LDARF1,R4      ; PTR TO TESTING ROUTINE
1662  007140  012705  006556            MOV     #LDAF1,R5       ; PTR TO TEST DATA SET
1663  007144  004737  021334            JSR     PC,@#LDAFT      ; GO TEST
1664
1665  007150  000400                    BR      TST75           ;;
1666
1667                            ; ( FOR DATA SEE TEST OF FP REG AC0 WITH TEST PATTERN LDAF-1, ABOVE )
1668
1669                            ;;*****************************************************************
1670                            ;*TEST 75       TEST FP AC1, F MODE, TEST PATTERN LDAF-2
1671                            ;*****************************************************************
1672  007152  000004            TST75:  SCOPE
1673  007154  012704  021536            MOV     #LDARF1,R4      ; PTR TO TESTING ROUTINE
1674  007160  012705  006630            MOV     #LDAF2,R5       ; PTR TO TEST DATA SET
1675  007164  004737  021334            JSR     PC,@#LDAFT      ; GO TEST
1676
1677  007170  000400                    BR      TST76           ;;
1678
1679                            ; ( FOR DATA SEE TEST OF FP REG AC0 WITH TEST PATTERN LDAF-2, ABOVE )
1680
1681                            ;;*****************************************************************
1682                            ;*TEST 76       TEST FP AC1, F MODE, TEST PATTERN LDAF-3
1683                            ;*****************************************************************
1684  007172  000004            TST76:  SCOPE
1685  007174  012704  021536            MOV     #LDARF1,R4      ; PTR TO TESTING ROUTINE
1686  007200  012705  006702            MOV     #LDAF3,R5       ; PTR TO TEST DATA SET
1687  007204  004737  021334            JSR     PC,@#LDAFT      ; GO TEST
1688
1689  007210  000400                    BR      TST77           ;;
1690
1691                            ; ( FOR DATA SEE TEST OF FP REG AC0 WITH TEST PATTERN LDAF-3, ABOVE )
1692
1693                            ;;*****************************************************************
1694                            ;*TEST 77       TEST FP AC1, F MODE, TEST PATTERN LDAF-4
1695                            ;*****************************************************************
1696  007212  000004            TST77:  SCOPE
1697  007214  012704  021536            MOV     #LDARF1,R4      ; PTR TO TESTING ROUTINE
1698  007220  012705  006754            MOV     #LDAF4,R5       ; PTR TO TEST DATA SET
1699  007224  004737  021334            JSR     PC,@#LDAFT      ; GO TEST
1700
1701  007230  000400                    BR      TST100          ;;
1702
1703                            ; ( FOR DATA SEE TEST OF FP REG AC0 WITH TEST PATTERN LDAF-4, ABOVE )
1704
1705                            ;;*****************************************************************
1706                            ;*TEST 100      TEST FP AC1, F MODE, TEST PATTERN LDAF-5
1707                            ;*****************************************************************
1708  007232  000004            TST100: SCOPE
1709  007234  012704  021536            MOV     #LDARF1,R4      ; PTR TO TESTING ROUTINE
```

```
1710  007240  012705  007026          T100    TEST FP AC1, F MODE, TEST PATTERN LDAF-5
1711  007244  004737  021334                  MOV     #LDAF5,R5       ; PTR TO TEST DATA SET
1712                                          JSR     PC,@#LDAFT      ; GO TEST
1713  007250  000400                         BR      TST101          ;;
1714
1715                                  ; ( FOR DATA SEE TEST OF FP REG AC0 WITH TEST PATTERN LDAF-5, ABOVE )
1716
1717                                  ;;*********************************************************************
1718                                  ;*TEST 101       TEST FP AC1, F MODE, TEST PATTERN LDAF-6
1719                                  ;;*********************************************************************
1720  007252  000004                 TST101: SCOPE
1721  007254  012704  021536                 MOV     #LDARF1,R4      ; PTR TO TESTING ROUTINE
1722  007260  012705  007100                 MOV     #LDAF6,R5       ; PTR TO TEST DATA SET
1723  007264  004737  021334                 JSR     PC,@#LDAFT      ; GO TEST
1724
1725  007270  000400                         BR      TST102          ;;
1726
1727                                  ; ( FOR DATA SEE TEST OF FP REG AC0 WITH TEST PATTERN LDAF-6, ABOVE )
1728
1729
1730                                  ;;*********************************************************************
1731                                  ;*TEST 102       TEST FP AC2, F MODE, TEST PATTERN LDAF-1
1732                                  ;;*********************************************************************
1733  007272  000004                 TST102: SCOPE
1734  007274  012704  021560                 MOV     #LDARF2,R4      ; PTR TO TESTING ROUTINE
1735  007300  012705  006556                 MOV     #LDAF1,R5       ; PTR TO TEST DATA SET
1736  007304  004737  021334                 JSR     PC,@#LDAFT      ; GO TEST
1737
1738  007310  000400                         BR      TST103          ;;
1739
1740                                  ; ( FOR DATA SEE TEST OF FP REG AC0 WITH TEST PATTERN LDAF-1, ABOVE )
1741
1742                                  ;;*********************************************************************
1743                                  ;*TEST 103       TEST FP AC2, F MODE, TEST PATTERN LDAF-2
1744                                  ;;*********************************************************************
1745  007312  000004                 TST103: SCOPE
1746  007314  012704  021560                 MOV     #LDARF2,R4      ; PTR TO TESTING ROUTINE
1747  007320  012705  006630                 MOV     #LDAF2,R5       ; PTR TO TEST DATA SET
1748  007324  004737  021334                 JSR     PC,@#LDAFT      ; GO TEST
1749
1750  007330  000400                         BR      TST104          ;;
1751
1752                                  ; ( FOR DATA SEE TEST OF FP REG AC0 WITH TEST PATTERN LDAF-2, ABOVE )
1753
1754                                  ;;*********************************************************************
1755                                  ;*TEST 104       TEST FP AC2, F MODE, TEST PATTERN LDAF-3
1756                                  ;;*********************************************************************
1757  007332  000004                 TST104: SCOPE
1758  007334  012704  021560                 MOV     #LDARF2,R4      ; PTR TO TESTING ROUTINE
1759  007340  012705  006702                 MOV     #LDAF3,R5       ; PTR TO TEST DATA SET
1760  007344  004737  021334                 JSR     PC,@#LDAFT      ; GO TEST
1761
1762  007350  000400                         BR      TST105          ;;
1763
1764                                  ; ( FOR DATA SEE TEST OF FP REG AC0 WITH TEST PATTERN LDAF-3, ABOVE )
1765
```

```
                                        ;;*****************************************************************
                                        ;*TEST 105       TEST FP AC2, F MODE, TEST PATTERN LDAF-4
                                        ;;*****************************************************************
007352  000004                  TST105: SCOPE
007354  012704  021560                  MOV     #LDARF2,R4      ; PTR TO TESTING ROUTINE
007360  012705  006754                  MOV     #LDAF4,R5       ; PTR TO TEST DATA SET
007364  004737  021334                  JSR     PC,@#LDAFT      ; GO TEST
007370  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
                                        ;;*****************************************************************
007372  000004                  TST106: SCOPE
007374  012704  021560                  MOV     #LDARF2,R4      ; PTR TO TESTING ROUTINE
007400  012705  007026                  MOV     #LDAF5,R5       ; PTR TO TEST DATA SET
007404  004737  021334                  JSR     PC,@#LDAFT      ; GO TEST
007410  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
                                        ;;*****************************************************************
007412  000004                  TST107: SCOPE
007414  012704  021560                  MOV     #LDARF2,R4      ; PTR TO TESTING ROUTINE
007420  012705  007100                  MOV     #LDAF6,R5       ; PTR TO TEST DATA SET
007424  004737  021334                  JSR     PC,@#LDAFT      ; GO TEST
007430  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
                                        ;;*****************************************************************
007432  000004                  TST110: SCOPE
007434  012704  021602                  MOV     #LDARF3,R4      ; PTR TO TESTING ROUTINE
007440  012705  006556                  MOV     #LDAF1,R5       ; PTR TO TEST DATA SET
007444  004737  021334                  JSR     PC,@#LDAFT      ; GO TEST
007450  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
                                        ;;*****************************************************************
007452  000004                  TST111: SCOPE
007454  012704  021602                  MOV     #LDARF3,R4      ; PTR TO TESTING ROUTINE
007460  012705  006630                  MOV     #LDAF2,R5       ; PTR TO TEST DATA SET
007464  004737  021334                  JSR     PC,@#LDAFT      ; GO TEST
```

```
1822
1823  007470  000400                        BR      TST112          ;;
1824
1825                                  ; ( FOR DATA SEE TEST OF FP REG AC0 WITH TEST PATTERN LDAF-2, ABOVE )
1826
1827                                  ;;*****************************************************************
1828                                  ;*TEST 112      TEST FP AC3, F MODE, TEST PATTERN LDAF-3
1829                                  ;;*****************************************************************
1830  007472  000004                  TST112: SCOPE
1831  007474  012704  021602                  MOV     #LDARF3,R4      ; PTR TO TESTING ROUTINE
1832  007500  012705  006702                  MOV     #LDAF3,R5       ; PTR TO TEST DATA SET
1833  007504  004737  021334                  JSR     PC,@#LDAFT      ; GO TEST
1834
1835  007510  000400                        BR      TST113          ;;
1836
1837                                  ; ( FOR DATA SEE TEST OF FP REG AC0 WITH TEST PATTERN LDAF-3, ABOVE )
1838
1839                                  ;;*****************************************************************
1840                                  ;*TEST 113      TEST FP AC3, F MODE, TEST PATTERN LDAF-4
1841                                  ;;*****************************************************************
1842  007512  000004                  TST113: SCOPE
1843  007514  012704  021602                  MOV     #LDARF3,R4      ; PTR TO TESTING ROUTINE
1844  007520  012705  006754                  MOV     #LDAF4,R5       ; PTR TO TEST DATA SET
1845  007524  004737  021334                  JSR     PC,@#LDAFT      ; GO TEST
1846
1847  007530  000400                        BR      TST114          ;;
1848
1849                                  ; ( FOR DATA SEE TEST OF FP REG AC0 WITH TEST PATTERN LDAF-4, ABOVE )
1850
1851                                  ;;*****************************************************************
1852                                  ;*TEST 114      TEST FP AC3, F MODE, TEST PATTERN LDAF-5
1853                                  ;;*****************************************************************
1854  007532  000004                  TST114: SCOPE
1855  007534  012704  021602                  MOV     #LDARF3,R4      ; PTR TO TESTING ROUTINE
1856  007540  012705  007026                  MOV     #LDAF5,R5       ; PTR TO TEST DATA SET
1857  007544  004737  021334                  JSR     PC,@#LDAFT      ; GO TEST
1858
1859  007550  000400                        BR      TST115          ;;
1860
1861                                  ; ( FOR DATA SEE TEST OF FP REG AC0 WITH TEST PATTERN LDAF-5, ABOVE )
1862
1863                                  ;;*****************************************************************
1864                                  ;*TEST 115      TEST FP AC3, F MODE, TEST PATTERN LDAF-6
1865                                  ;;*****************************************************************
1866  007552  000004                  TST115: SCOPE
1867  007554  012704  021602                  MOV     #LDARF3,R4      ; PTR TO TESTING ROUTINE
1868  007560  012705  007100                  MOV     #LDAF6,R5       ; PTR TO TEST DATA SET
1869  007564  004737  021334                  JSR     PC,@#LDAFT      ; GO TEST
1870
1871  007570  000400                        BR      TST116          ;;
1872
1873                                  ; ( FOR DATA SEE TEST OF FP REG AC0 WITH TEST PATTERN LDAF-6, ABOVE )
1874
1875
1876                                  ;;*****************************************************************
1877                                  ;*TEST 116      TEST FP AC4, F MODE, TEST PATTERN LDAF-1
```

```
                                ;;*****************************************************************
007572  000004                 TST116: SCOPE
007574  012704  021624                 MOV     #LDARF4,R4      ; PTR TO TESTING ROUTINE
007600  012705  006556                 MOV     #LDAF1,R5       ; PTR TO TEST DATA SET
007604  004737  021334                 JSR     PC,@#LDAFT      ; GO TEST

007610  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
                                ;;*****************************************************************
007612  000004                 TST117: SCOPE
007614  012704  021624                 MOV     #LDARF4,R4      ; PTR TO TESTING ROUTINE
007620  012705  006630                 MOV     #LDAF2,R5       ; PTR TO TEST DATA SET
007624  004737  021334                 JSR     PC,@#LDAFT      ; GO TEST

007630  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
                                ;;*****************************************************************
007632  000004                 TST120: SCOPE
007634  012704  021624                 MOV     #LDARF4,R4      ; PTR TO TESTING ROUTINE
007640  012705  006702                 MOV     #LDAF3,R5       ; PTR TO TEST DATA SET
007644  004737  021334                 JSR     PC,@#LDAFT      ; GO TEST

007650  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
                                ;;*****************************************************************
007652  000004                 TST121: SCOPE
007654  012704  021624                 MOV     #LDARF4,R4      ; PTR TO TESTING ROUTINE
007660  012705  006754                 MOV     #LDAF4,R5       ; PTR TO TEST DATA SET
007664  004737  021334                 JSR     PC,@#LDAFT      ; GO TEST

007670  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
                                ;;*****************************************************************
007672  000004                 TST122: SCOPE
007674  012704  021624                 MOV     #LDARF4,R4      ; PTR TO TESTING ROUTINE
007700  012705  007026                 MOV     #LDAF5,R5       ; PTR TO TEST DATA SET
007704  004737  021334                 JSR     PC,@#LDAFT      ; GO TEST

007710  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
                                        ;;*****************************************************************
007712  000004                          TST123: SCOPE
007714  012704  021654                          MOV     #LDARF5,R4      ; PTR TO TESTING ROUTINE
007720  012705  006556                          MOV     #LDAF1,R5       ; PTR TO TEST DATA SET
007724  004737  021334                          JSR     PC,@#LDAFT      ; GO TEST
007730  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
                                        ;;*****************************************************************
007732  000004                          TST124: SCOPE
007734  012704  021654                          MOV     #LDARF5,R4      ; PTR TO TESTING ROUTINE
007740  012705  006630                          MOV     #LDAF2,R5       ; PTR TO TEST DATA SET
007744  004737  021334                          JSR     PC,@#LDAFT      ; GO TEST
007750  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
                                        ;;*****************************************************************
007752  000004                          TST125: SCOPE
007754  012704  021654                          MOV     #LDARF5,R4      ; PTR TO TESTING ROUTINE
007760  012705  006702                          MOV     #LDAF3,R5       ; PTR TO TEST DATA SET
007764  004737  021334                          JSR     PC,@#LDAFT      ; GO TEST

007770  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
                                        ;;*****************************************************************
007772  000004                          TST126: SCOPE
007774  012704  021654                          MOV     #LDARF5,R4      ; PTR TO TESTING ROUTINE
010000  012705  006754                          MOV     #LDAF4,R5       ; PTR TO TEST DATA SET
010004  004737  021334                          JSR     PC,@#LDAFT      ; GO TEST

010010  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
                                        ;;*****************************************************************
010012  000004                          TST127: SCOPE
010014  012704  021654                          MOV     #LDARF5,R4      ; PTR TO TESTING ROUTINE
```

```
1990  010020  012705  007026          MOV     #LDAF5,R5       ; PTR TO TEST DATA SET
1991  010024  004737  021334          JSR     PC,@#LDAF:      ; GO TEST
1992
1993  010030  000400                  BR      TST130          ;;
1994
1995                                  ; ( FOR DATA SEE TEST OF FP REG AC0 WITH TEST PATTERN LDAF-5, ABOVE )
1996
1997
1998
```

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

                                ;;*****************************************************************
                                ;*TEST 130      TEST OF LOAD-FSRC MODE-1, STORE-FDST MODE-6(PC), D MODE
                                ;;*****************************************************************
010032  000004                  TST130: SCOPE
010034  170127  047600                  LDFPS   #047600         ; D MODE FPS
010040  016767  000102  171122          MOV     ADRD1+0,$REG0   ; MOVE PATTERN
010046  016767  000076  171116          MOV     ADRD1+2,$REG1   ;
010054  016767  000072  171112          MOV     ADRD1+4,$REG2   ;
010062  016767  000066  171106          MOV     ADRD1+6,$REG3   ;

010070  012705  001170                  MOV     #$REG0,R5       ; ADDR(DATA)
010074  172415                          LDD     (R5),AC0        ; M1-R5

010076  174067  171076                  STD     AC0,$REG4       ; M6-R7

010102  026767  171062  171070          CMP     $REG0,$REG4     ; 1ST WORD PATTERN CHECK?
010110  001014                          BNE     64$             ;
010112  026767  171054  171062          CMP     $REG1,$REG5     ; 2ND WORD PATTERN CHECK?
010120  001010                          BNE     64$             ;
010122  026767  171046  171054          CMP     $REG2,$REG6     ; 3RD WORD PATTERN CHECK?
010130  001004                          BNE     64$             ;
010132  026767  171040  171046          CMP     $REG3,$REG7     ; 4TH WORD PATTERN CHECK?
010140  001401                          BEQ     65$             ;
010142  104031                  64$:    ERROR   31              ; PATTERN DOESNT MATCH

010144                          65$:
010144  000404                          BR      TST131          ;;

010146  052525  052525  052525  ADRD1:  .WORD   ALTP,ALTP,ALTP,ALTP     ; TEST PATTERN
010154  052525

                                ;;*****************************************************************
                                ;*TEST 131      TEST OF LOAD-FSRC MODE-4, STORE-FDST MODE-3(PC), D MODE
                                ;;*****************************************************************
010156  000004                  TST131: SCOPE
010160  170127  047600                  LDFPS   #047600         ; D MODE FPS
010164  016767  000112  170776          MOV     ADRD2+0,$REG0   ; MOVE PATTERN
010172  016767  000106  170772          MOV     ADRD2+2,$REG1   ;
010200  016767  000102  170766          MOV     ADRD2+4,$REG2   ;
010206  016767  000076  170762          MOV     ADRD2+6,$REG3   ;

010214  012704  001200                  MOV     #$REG0+10,R4    ; ADDR(DATA+10)
010220  172544                          LDD     -(R4),AC1       ; M4-R4

010222  174137  001200                  STD     AC1,@#$REG4     ; M3-R7

010226  020427  001170                  CMP     R4,#$REG0       ; WAS FSRC ADDR REG DECRE RIGHT AMOUNT?
010232  001401                          BEQ     66$             ;
010234  104032                          ERROR   32              ; NOT EQUAL, SIGNAL ERROR
010236                          66$:
```

```
2055
2056  010236  026767  170726  170734          CMP     $REG0,$REG4     ; 1ST WORD PATTERN CHECK?
2057  010244  001014                          BNE     64$             ;
2058  010246  026767  170720  170726          CMP     $REG1,$REG5     ; 2ND WORD PATTERN CHECK?
2059  010254  001010                          BNE     64$             ;
2060  010256  026767  170712  170720          CMP     $REG2,$REG6     ; 3RD WORD PATTERN CHECK?
2061  010264  001004                          BNE     64$             ;
2062  010266  026767  170704  170712          CMP     $REG3,$REG7     ; 4TH WORD PATTERN CHECK?
2063  010274  001401                          BEQ     65$             ;
2064  010276  104033                  64$:    ERROR   33              ; PATTERN DOESNT MATCH
2065
2066  010300                          65$:
2067  010300  000404                          BR      TST132          ;;
2068
2069  010302  170360  170360  170360  ADRD2:  .WORD   ALT4N,ALT4N,ALT4N,ALT4N ; TEST PATTERN
2070  010310  170360
2071
2072
2073                                  ;;*****************************************************************
2074                                  ;*TEST 132      TEST OF LOAD-FSRC MODE-7, STORE-FDST MODE-3, D MODE
2075                                  ;;*****************************************************************
2076  010312  000004                  TST132: SCOPE
2077  010314  170127  047600                  LDFPS   #047600         ; D MODE FPS
2078  010320  016767  000132  170642          MOV     ADRD3+0,$REG0   ; MOVE PATTERN
2079  010326  016767  000126  170636          MOV     ADRD3+2,$REG1   ;
2080  010334  016767  000122  170632          MOV     ADRD3+4,$REG2   ;
2081  010342  016767  000116  170626          MOV     ADRD3+6,$REG3   ;
2082
2083  010350  012767  001170  170632          MOV     #$REG0,$REG10   ; ADDR(DATA)
2084  010356  012705  001214                  MOV     #$REG10+4,R5    ; ADDR(ADDR(DATA))+4
2085  010362  172675  177774                  LDD     @-4(R5),AC2     ; M7-R5
2086
2087  010366  012767  001200  170616          MOV     #$REG4,$REG11   ; ADDR(DEST)
2088  010374  012704  001212                  MOV     #$REG11,R4      ; ADDR(ADDR(DEST))
2089  010400  174234                          STD     AC2,@(R4)+      ; M3-R4
2090
2091  010402  020427  001214                  CMP     R4,#$REG11+2    ; WAS FDST ADDR REG INCRE RIGHT AMOUNT?
2092  010406  001401                          BEQ     66$             ;
2093  010410  104034                          ERROR   34              ; NOT EQUAL, SIGNAL ERROR
2094  010412                          66$:
2095
2096  010412  026767  170552  170560          CMP     $REG0,$REG4     ; 1ST WORD PATTERN CHECK?
2097  010420  001014                          BNE     64$             ;
2098  010422  026767  170544  170552          CMP     $REG1,$REG5     ; 2ND WORD PATTERN CHECK?
2099  010430  001010                          BNE     64$             ;
2100  010432  026767  170536  170544          CMP     $REG2,$REG6     ; 3RD WORD PATTERN CHECK?
2101  010440  001004                          BNE     64$             ;
2102  010442  026767  170530  170536          CMP     $REG3,$REG7     ; 4TH WORD PATTERN CHECK?
2103  010450  001401                          BEQ     65$             ;
2104  010452  104035                  64$:    ERROR   35              ; PATTERN DOESNT MATCH
2105
2106  010454                          65$:
2107  010454  000404                          BR      TST133          ;;
2108
2109  010456  125252  125252  125252  ADRD3:  .WORD   ALTN,ALTN,ALTN,ALTN     ; TEST PATTERN
2110  010464  125252
```

```
                                ;;*****************************************************************
                                ;*TEST 133      TEST OF LOAD-FSRC MODE-6(PC), STORE-FDST MODE-2, D MODE
                                ;;*****************************************************************
010466  000004                  TST133: SCOPE
010470  170127  047600                  LDFPS   #047600         ; D MODE FPS
010474  016767  000112  170466          MOV     ADRD4+0,$REG0   ; MOVE PATTERN
010502  016767  000106  170462          MOV     ADRD4+2,$REG1   ;
010510  016767  000102  170456          MOV     ADRD4+4,$REG2   ;
010516  016767  000076  170452          MOV     ADRD4+6,$REG3   ;

010524  172767  170440                  LDD     $REG0,AC3       ; M6-R7

010530  012705  001200                  MOV     #$REG4,R5       ; ADDR(DEST)
010534  174325                          STD     AC3,(R5)+       ; M2-R5

010536  020527  001210                  CMP     R5,#$REG4+10    ; WAS FDST ADDR REG INCRE RIGHT AMOUNT?
010542  001401                          BEQ     66$             ;
010544  104036                          ERROR   36              ; NOT EQUAL, SIGNAL ERROR
010546                          66$:

010546  026767  170416  170424          CMP     $REG0,$REG4     ; 1ST WORD PATTERN CHECK?
010554  001014                          BNE     64$             ;
010556  026767  170410  170416          CMP     $REG1,$REG5     ; 2ND WORD PATTERN CHECK?
010564  001010                          BNE     64$             ;
010566  026767  170402  170410          CMP     $REG2,$REG6     ; 3RD WORD PATTERN CHECK?
010574  001004                          BNE     64$             ;
010576  026767  170374  170402          CMP     $REG3,$REG7     ; 4TH WORD PATTERN CHECK?
010604  001401                          BEQ     65$             ;
010606  104031                  64$:    ERROR   31              ; PATTERN DOESNT MATCH

010610                          65$:
010610  000404                          BR      TST134          ;;

010612  007417  007417  007417  ADRD4:  .WORD   ALT4P,ALT4P,ALT4P,ALT4P ; TEST PATTERN
010620  007417

                                ;;*****************************************************************
                                ;*TEST 134      TEST OF LOAD-FSRC MODE-2, STORE-FDST MODE-1, D MODE
                                ;;*****************************************************************
010622  000004                  TST134: SCOPE
010624  170127  047600                  LDFPS   #047600         ; D MODE FPS
010630  016767  000114  170332          MOV     ADRD5+0,$REG0   ; MOVE PATTERN
010636  016767  000110  170326          MOV     ADRD5+2,$REG1   ;
010644  016767  000104  170322          MOV     ADRD5+4,$REG2   ;
010652  016767  000100  170316          MOV     ADRD5+6,$REG3   ;

010660  012700  001170                  MOV     #$REG0,R0       ; ADDR(DATA)
010664  172420                          LDD     (R0)+,AC0       ; M1-R0

010666  012703  001200                  MOV     #$REG4,R3       ; ADDR(DEST)
010672  174013                          STD     AC0,(R3)        ; M1-R3

010674  020027  001200                  CMP     R0,#$REG0+10    ; WAS FSRC ADDR REG INCRE RIGHT AMOUNT?
```

```
2167  010700  001401                          BEQ     66$             ;
2168  010702  104037                          ERROR   37              ; NOT EQUAL, SIGNAL ERROR
2169  010704                          66$:
2170
2171  010704  026767  170260  170266          CMP     $REG0,$REG4     ; 1ST WORD PATTERN CHECK?
2172  010712  001014                          BNE     64$             ;
2173  010714  026767  170252  170260          CMP     $REG1,$REG5     ; 2ND WORD PATTERN CHECK?
2174  010722  001010                          BNE     64$             ;
2175  010724  026767  170244  170252          CMP     $REG2,$REG6     ; 3RD WORD PATTERN CHECK?
2176  010732  001004                          BNE     64$             ;
2177  010734  026767  170236  170244          CMP     $REG3,$REG7     ; 4TH WORD PATTERN CHECK?
2178  010742  001401                          BEQ     65$             ;
2179  010744  104040                  64$:    ERROR   40              ; PATTERN DOESNT MATCH
2180
2181  010746                          65$:
2182  010746  000404                          BR      TST135          ;;
2183
2184  010750  125252  125252  125252  ADR05:  .WORD   ALTN,ALTN,ALTN,ALTN    ; TEST PATTERN
2185  010756  125252
2186
2187
2188                                  ;;*****************************************************************
2189                                  ;*TEST 135       TEST OF LOAD-FSRC MODE-5, STORE-FDST MODE-7(PC), D MODE
2190                                  ;;*****************************************************************
2191  010760  000004                  TST135: SCOPE
2192  010762  170127  047600                  LDFPS   #047600         ; D MODE FPS
2193  010766  016767  000126  170174          MOV     ADR06+0,$REG0   ; MOVE PATTERN
2194  010774  016767  000122  170170          MOV     ADR06+2,$REG1   ;
2195  011002  016767  000116  170164          MOV     ADR06+4,$REG2   ;
2196  011010  016767  000112  170160          MOV     ADR06+6,$REG3   ;
2197
2198  011016  012767  001170  170164          MOV     #$REG0,$REG10   ; ADDR(DATA)
2199  011024  012701  001212                  MOV     #$REG10+2,R1    ; ADDR(ADDR(DATA)+2)
2200  011030  172551                          LDD     @-(R1),AC1      ; M5-R1
2201
2202  011032  012767  001200  170152          MOV     #$REG4,$REG11   ; ADDR(DEST)
2203  011040  174177  170146                  STD     AC1,@$REG11     ; M7-R7
2204
2205  011044  020127  001210                  CMP     R1,#$REG10      ; WAS FSRC ADDR REG DECRE RIGHT AMOUNT?
2206  011050  001401                          BEQ     66$             ;
2207  011052  104041                          ERROR   41              ; NOT EQUAL, SIGNAL ERROR
2208  011054                          66$:
2209
2210  011054  026767  170110  170116          CMP     $REG0,$REG4     ; 1ST WORD PATTERN CHECK?
2211  011062  001014                          BNE     64$             ;
2212  011064  026767  170102  170110          CMP     $REG1,$REG5     ; 2ND WORD PATTERN CHECK?
2213  011072  001010                          BNE     64$             ;
2214  011074  026767  170074  170102          CMP     $REG2,$REG6     ; 3RD WORD PATTERN CHECK?
2215  011102  001004                          BNE     64$             ;
2216  011104  026767  170066  170074          CMP     $REG3,$REG7     ; 4TH WORD PATTERN CHECK?
2217  011112  001401                          BEQ     65$             ;
2218  011114  104042                  64$:    ERROR   42              ; PATTERN DOESNT MATCH
2219
2220  011116                          65$:
2221  011116  000404                          BR      TST136          ;;
2222
```

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

```
2279  011352  012704  001212                    MOV     #$REG4+12,R4    ; ADDR(DEST)+12
2280  011356  174364  177766                    STD     AC3,-12(R4)     ; M6-R4
2281
2282  011362  026767  167602  167610            CMP     $REG0,$REG4     ; 1ST WORD PATTERN CHECK?
2283  011370  001014                            BNE     64$             ;
2284  011372  026767  167574  167602            CMP     $REG1,$REG5     ; 2ND WORD PATTERN CHECK?
2285  011400  001010                            BNE     64$             ;
2286  011402  026767  167566  167574            CMP     $REG2,$REG6     ; 3RD WORD PATTERN CHECK?
2287  011410  001004                            BNE     64$             ;
2288  011412  026767  167560  167566            CMP     $REG3,$REG7     ; 4TH WORD PATTERN CHECK?
2289  011420  001401                            BEQ     65$             ;
2290  011422  104044                    64$:    ERROR   44              ; PATTERN DOESNT MATCH
2291
2292  011424                            65$:
2293  011424  000404                            BR      TST140          ;;
2294
2295  011426  170360  170360  170360  ADRD10: .WORD   ALT4N,ALT4N,ALT4N,ALT4N ; TEST PATTERN
2296  011434  170360
2297
2298
2299                                    ;;*****************************************************************
2300                                    ;*TEST 140      TEST OF LOAD-FSRC MODE-3, STORE-FDST MODE-2(PC), D MODE
2301                                    ;;*****************************************************************
2302  011436  000004                    TST140: SCOPE
2303  011440  170127  047600                    LDFPS   #047600         ; D MODE FPS
2304  011444  016767  000114  167516            MOV     ADRD11+0,$REG0  ; MOVE PATTERN
2305  011452  016767  000110  167512            MOV     ADRD11+2,$REG1  ;
2306  011460  016767  000104  167506            MOV     ADRD11+4,$REG2  ;
2307  011466  016767  000100  167502            MOV     ADRD11+6,$REG3  ;
2308
2309  011474  012767  001170  167506            MOV     #$REG0,$REG10   ; ADDR(DATA)
2310  011502  012702  001210                    MOV     #$REG10,R2      ; ADDR(ADDR(DATA))
2311  011506  172432                            LDD     @(R2)+,AC0      ; M3-R2
2312
2313  011510  012767  104117  000002            MOV     #ERROR117,1$    ; SETUP ERROR CALL FOR WRONG INCREMENT
2314  011516  174027                            STD     AC0,(PC)+       ; M2-R7
2315  011520  000000                    1$:     .WORD   0               ; DEST
2316  011522  000403                            BR      2$              ; THIS SHOULD BE NEXT INSTR AFTER FPP PC MODE 2
2317  011524  104120                            ERROR   120             ; NOT HERE  (+4 INCRE)
2318  011526  104121                            ERROR   121             ; OR HERE  (+6 INCRE)
2319  011530  104122                            ERROR   122             ; OR HERE  (+10 INCRE)
2320  011532  016767  177762  167440    2$:     MOV     1$,$REG4        ; STORE DEST
2321
2322  011540  020227  001212                    CMP     R2,#$REG10+2    ; WAS FSRC ADDR REG INCRE BY RIGHT AMOUNT?
2323  011544  001401                            BEQ     64$             ;
2324  011546  104045                            ERROR   45              ; NOT EQUAL, SIGNAL ERROR
2325  011550                            64$:
2326
2327  011550  026767  167414  167422            CMP     $REG0,$REG4     ; WAS 1 WORD STORED OK?
2328  011556  001401                            BEQ     65$             ;
2329  011560  104046                            ERROR   46              ; NOT EQUAL, SIGNAL ERROR
2330  011562                            65$:
2331
2332  011562  000404                            BR      TST141          ;;
2333
2334  011564  052525  000000  000000  ADRD11: .WORD   ALTP,0,0,0        ; TEST PATTERN
```

```
2335  011572  000000
2336
2337
2338                                  ;;*****************************************************************
2339                                  ;*TEST 141      TEST OF LOAD-FSRC MODE-6, STORE-FDST MODE-4, D MODE
2340                                  ;*****************************************************************
2341  011574  000004                 TST141: SCOPE
2342  011576  170127  047600                 LDFPS   #047600         ; D MODE FPS
2343  011602  016767  000116  167360        MOV     ADRD12+0,$REG0  ; MOVE PATTERN
2344  011610  016767  000112  167354        MOV     ADRD12+2,$REG1  ;
2345  011616  016767  000106  167350        MOV     ADRD12+4,$REG2  ;
2346  011624  016767  000102  167344        MOV     ADRD12+6,$REG3  ;
2347
2348  011632  012703  001262                 MOV     #$REG0+72,R3    ; ADDR(DATA)+72
2349  011636  172563  177706                 LDD     -72(R3),AC1     ; M6-R3
2350
2351  011642  012701  001210                 MOV     #$REG4+10,R1    ; ADDR(DEST+10)
2352  011646  174141                         STD     AC1,-(R1)       ; M4-R1
2353
2354  011650  020127  001200                 CMP     R1,#$REG4       ; WAS FDST ADDR REG DECRE RIGHT AMOUNT?
2355  011654  001401                         BEQ     66$             ;
2356  011656  104047                         ERROR   47              ; NOT EQUAL, SIGNAL ERROR
2357  011660                         66$:
2358
2359  011660  026767  167304  167312         CMP     $REG0,$REG4     ; 1ST WORD PATTERN CHECK?
2360  011666  001014                         BNE     64$             ;
2361  011670  026767  167276  167304         CMP     $REG1,$REG5     ; 2ND WORD PATTERN CHECK?
2362  011676  001010                         BNE     64$             ;
2363  011700  026767  167270  167276         CMP     $REG2,$REG6     ; 3RD WORD PATTERN CHECK?
2364  011706  001004                         BNE     64$             ;
2365  011710  026767  167262  167270         CMP     $REG3,$REG7     ; 4TH WORD PATTERN CHECK?
2366  011716  001401                         BEQ     65$             ;
2367  011720  104050                 64$:    ERROR   50              ; PATTERN DOESNT MATCH
2368
2369  011722                         65$:
2370  011722  000404                         BR      TST142          ;;
2371
2372  011724  170360  170360  170360 ADRD12: .WORD   ALT4N,ALT4N,ALT4N,ALT4N ; TEST PATTERN
2373  011732  170360
2374
2375
2376                                  ;;*****************************************************************
2377                                  ;*TEST 142      TEST OF LOAD-FSRC MODE-3(PC), STORE-FDST MODE-0, D MODE
2378                                  ;*****************************************************************
2379  011734  000004                 TST142: SCOPE
2380  011736  170127  047600                 LDFPS   #047600         ; D MODE FPS
2381  011742  016767  000102  167220        MOV     ADRD13+0,$REG0  ; MOVE PATTERN
2382  011750  016767  000076  167214        MOV     ADRD13+2,$REG1  ;
2383  011756  016767  000072  167210        MOV     ADRD13+4,$REG2  ;
2384  011764  016767  000066  167204        MOV     ADRD13+6,$REG3  ;
2385
2386  011772  172637  001170                 LDD     @#$REG0,AC2     ; M3-R7
2387
2388  011776  174203                         STD     AC2,AC3         ; M0-R3
2389  012000  174367  167174                 STD     AC3,$REG4       ; M6-R7 WORKS
2390
```

```
2391  012004  026767  167160  167166          CMP     $REG0,$REG4     ; 1ST WORD PATTERN CHECK?
2392  012012  001014                          BNE     64$             ;
2393  012014  026767  167152  167160          CMP     $REG1,$REG5     ; 2ND WORD PATTERN CHECK?
2394  012022  001010                          BNE     64$             ;
2395  012024  026767  167144  167152          CMP     $REG2,$REG6     ; 3RD WORD PATTERN CHECK?
2396  012032  001004                          BNE     64$             ;
2397  012034  026767  167136  167144          CMP     $REG3,$REG7     ; 4TH WORD PATTERN CHECK?
2398  012042  001401                          BEQ     65$             ;
2399  012044  104051                  64$:    ERROR   51              ; PATTERN DOESNT MATCH
2400
2401  012046                          65$:
2402  012046  000404                          BR      TST143          ;;
2403
2404  012050  125252  125252  125252  ADRD13: .WORD   ALTN,ALTN,ALTN,ALTN     ; TEST PATTERN
2405  012056  125252
2406
2407
2408                                  ;;*****************************************************************
2409                                  ;*TEST 143      TEST OF LOAD-FSRC MODE-0, STORE-FDST MODE-5, D MODE
2410                                  ;;*****************************************************************
2411  012060  000004                  TST143: SCOPE
2412  012062  170127  047600                  LDFPS   #047600         ; D MODE FPS
2413  012066  016767  000122  167074          MOV     ADRD14+0,$REG0  ; MOVE PATTERN
2414  012074  016767  000116  167070          MOV     ADRD14+2,$REG1  ;
2415  012102  016767  000112  167064          MOV     ADRD14+4,$REG2  ;
2416  012110  016767  000106  167060          MOV     ADRD14+6,$REG3  ;
2417
2418  012116  172467  167046                  LDD     $REG0,AC0       ; M6-R7 WORKS
2419  012122  172500                          LDD     AC0,AC1         ; M0-R0
2420
2421  012124  012767  001200  167056          MOV     #$REG4,$REG10   ; ADDR(DEST)
2422  012132  012702  001212                  MOV     #$REG10+2,R2    ; ADDR(ADDR(DEST)+2)
2423  012136  174152                          STD     AC1,@-(R2)      ; M5-R2
2424
2425  012140  020227  001210                  CMP     R2,#$REG10      ; WAS FDST ADDR REG DECRE RIGHT AMOUN'?
2426  012144  001401                          BEQ     66$             ;
2427  012146  104052                          ERROR   52              ; NOT EQUAL, SIGNAL ERROR
2428  012150                          66$:
2429
2430  012150  026767  167014  167022          CMP     $REG0,$REG4     ; 1ST WORD PATTERN CHECK?
2431  012156  001014                          BNE     64$             ;
2432  012160  026767  167006  167014          CMP     $REG1,$REG5     ; 2ND WORD PATTERN CHECK?
2433  012166  001010                          BNE     64$             ;
2434  012170  026767  167000  167006          CMP     $REG2,$REG6     ; 3RD WORD PATTERN CHECK?
2435  012176  001004                          BNE     64$             ;
2436  012200  026767  166772  167000          CMP     $REG3,$REG7     ; 4TH WORD PATTERN CHECK?
2437  012206  001401                          BEQ     65$             ;
2438  012210  104046                  64$:    ERROR   46              ; PATTERN DOESNT MATCH
2439
2440  012212                          65$:
2441  012212  000404                          BR      TST144          ::
2442
2443  012214  125252  125252  125252  ADRD14: .WORD   ALTN,ALTN,ALTN,ALTN     ; TEST PATTERN
2444  012222  125252
2445
2446
```

```
                                        ;;*****************************************************************
                                        ;*TEST 144      TEST OF LOAD-FSRC MODE-1, STORE-FDST MODE-4, F MODE
                                        ;;*****************************************************************
012224  000004                  TST144: SCOPE
012226  170127  047400                  LDFPS   #047400         ; F MODE FPS
012232  016767  000060  166730          MOV     ADRF1+0,$REG0   ; MOVE PATTERN
012240  016767  000054  166724          MOV     ADRF1+2,$REG1   ;

012246  012702  001170                  MOV     #$REG0,R2       ; ADDR(DATA)
012252  172412                          LDF     (R2),AC0        ; M1-R2

012254  012705  001200                  MOV     #$REG2+4,R5     ; ADDR(DEST+4)
012260  174045                          STF     AC0,-(R5)       ; M4-R5

012262  020527  001174                  CMP     R5,#$REG2       ; WAS FDST ADDR REG DECRE RIGHT AMOUNT?
012266  001401                          BEQ     66$             ;
012270  104053                          ERROR   53              ; NOT EQUAL, SIGNAL ERROR
012272                          66$:

012272  026767  166672  166674          CMP     $REG0,$REG2     ; 1ST WORD PATTERN CHECK?
012300  001004                          BNE     64$             ;
012302  026767  166664  166666          CMP     $REG1,$REG3     ; 2ND WORD PATTERN CHECK?
012310  001401                          BEQ     65$             ;
012312  104054                  64$:    ERROR   54              ; PATTERN DOESNT MATCH

012314                          65$:
012314  000402                          BR      TST145          ;;

012316  170360  170360          ADRF1:  .WORD   ALT4N,ALT4N     ; TEST PATTERN


                                        ;;*****************************************************************
                                        ;*TEST 145      TEST OF LOAD-FSRC MODE-3, STORE-FDST MODE-6, F MODE
                                        ;;*****************************************************************
012322  000004                  TST145: SCOPE
012324  170127  047400                  LDFPS   #047400         ; F MODE FPS
012330  016767  000070  166632          MOV     ADRF2+0,$REG0   ; MOVE PATTERN
012336  016767  000064  166626          MOV     ADRF2+2,$REG1   ;

012344  012767  001170  166626          MOV     #$REG0,$REG4    ; ADDR(DATA)
012352  012704  001200                  MOV     #$REG4,R4       ; ADDR(ADDR(DATA))
012356  172534                          LDF     @(R4)+,AC1      ; M3-R4

012360  012700  001166                  MOV     #$REG2-6,R0     ; ADDR(DEST)-6
012364  174160  000006                  STF     AC1,6(R0)       ; M6-R0

012370  020427  001202                  CMP     R4,#$REG4+2     ; WAS FSRC ADDR REG INCRE RIGHT AMOUNT?
012374  001401                          BEQ     66$             ;
012376  104055                          ERROR   55              ; NOT EQUAL, SIGNAL ERROR
012400                          66$:

012400  026767  166564  166566          CMP     $REG0,$REG2     ; 1ST WORD PATTERN CHECK?
012406  001004                          BNE     64$             ;
012410  026767  166556  166560          CMP     $REG1,$REG3     ; 2ND WORD PATTERN CHECK?
012416  001401                          BEQ     65$             ;
012420  104056                  64$:    ERROR   56              ; PATTERN DOESNT MATCH
```

```
2503
2504  012422                          65$:
2505  012422  000402                          BR      TST146          ;;
2506
2507  012424  052525  052525          ADRF2:  .WORD   ALTP,ALTP       ; TEST PATTERN
2508
2509
2510                                  ;;*****************************************************************
2511                                  ;*TEST 146      TEST OF LOAD-FSRC MODE-5, STORE-FDST MODE-3, F MODE
2512                                  ;;*****************************************************************
2513  012430  000004                  TST146: SCOPE
2514  012432  170127  047400                  LDFPS   #047400         ; F MODE FPS
2515  012436  016767  000104  166524          MOV     ADRF3+0,$REG0   ; MOVE PATTERN
2516  012444  016767  000100  166520          MOV     ADRF3+2,$REG1   ;
2517
2518  012452  012767  001170  166520          MOV     #$REG0,$REG4    ; ADDR(DATA)
2519  012460  012700  001202                  MOV     #$REG4+2,R0     ; ADDR(ADDR(DATA)+2)
2520  012464  172650                          LDF     @-(R0),AC2      ; M5-R0
2521
2522  012466  012767  001174  166506          MOV     #$REG2,$REG5    ; ADDR(DEST)
2523  012474  012702  001202                  MOV     #$REG5,R2       ; ADDR(ADDR(DEST))
2524  012500  174232                          STF     AC2,@(R2)+      ; M3-R2
2525
2526  012502  020027  001200                  CMP     R0,#$REG4       ; WAS FSRC ADDR REG DECRE RIGHT AMOUNT?
2527  012506  001401                          BEQ     66$             ;
2528  012510  104057                          ERROR   57              ; NOT EQUAL, SIGNAL ERROR
2529  012512                          66$:
2530
2531  012512  020227  001204                  CMP     R2,#$REG5+2     ; WAS FDST ADDR REG INCRE RIGHT AMOUNT?
2532  012516  001401                          BEQ     67$             ;
2533  012520  104060                          ERROR   60              ; NOT EQUAL, SIGNAL ERROR
2534  012522                          67$:
2535
2536  012522  026767  166442  166444          CMP     $REG0,$REG2     ; 1ST WORD PATTERN CHECK?
2537  012530  001004                          BNE     64$             ;
2538  012532  026767  166434  166436          CMP     $REG1,$REG3     ; 2ND WORD PATTERN CHECK?
2539  012540  001401                          BEQ     65$             ;
2540  012542  104061                  64$:    ERROR   61              ; PATTERN DOESNT MATCH
2541
2542  012544                          65$:
2543  012544  000402                          BR      TST147          ;;
2544
2545  012546  007417  007417          ADRF3:  .WORD   ALT4P,ALT4P     ; TEST PATTERN
2546
2547
2548                                  ;;*****************************************************************
2549                                  ;*TEST 147      TEST OF LOAD-FSRC MODE-7, STORE-FDST MODE-1, F MODE
2550                                  ;;*****************************************************************
2551  012552  000004                  TST147: SCOPE
2552  012554  170127  047400                  LDFPS   #047400         ; F MODE FPS
2553  012560  016767  000060  166402          MOV     ADRF4+0,$REG0   ; MOVE PATTERN
2554  012566  016767  000054  166376          MOV     ADRF4+2,$REG1   ;
2555
2556  012574  012767  001170  166376          MOV     #$REG0,$REG4    ; ADDR(DATA)
2557  012602  012702  001162                  MOV     #$REG4-16,R2    ; ADDR(ADDR(DATA))-16
2558  012606  172772  000016                  LDF     @16(R2),AC3     ; M7-R2
```

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

```
2615
2616  013000  012767  001174  166174          MOV     #$REG2,$REG5    ; ADDR(DEST)
2617  013006  012705  001226                  MOV     #$REG5+24,R5    ; ADDR(ADDR(DEST))+24
2618  013012  174175  177754                  STF     AC1,@-24(R5)    ; M7-R5
2619
2620  013016  026767  166146  166150          CMP     $REG0,$REG2     ; 1ST WORD PATTERN CHECK?
2621  013024  001004                          BNE     64$             ;
2622  013026  026767  166140  166142          CMP     $REG1,$REG3     ; 2ND WORD PATTERN CHECK?
2623  013034  001401                          BEQ     65$             ;
2624  013036  104065                  64$:    ERROR   65              ; PATTERN DOESNT MATCH
2625
2626  013040                          65$:
2627  013040  000402                          BR      TST152          ;;
2628
2629  013042  125252  125252          ADRF6:  .WORD   ALTN,ALTN       ; TEST PATTERN
2630
2631
2632                                  ;;*****************************************************************
2633                                  ;*TEST 152       TEST OF LOAD-FSRC MODE-2, STORE-FDST MODE-3(PC), F MODE
2634                                  ;;*****************************************************************
2635  013046  000004                  TST152: SCOPE
2636  013050  170127  047400                  LDFPS   #047400         ; F MODE FPS
2637  013054  016767  000056  166106          MOV     ADRF7+0,$REG0   ; MOVE PATTERN
2638  013062  016767  000052  166102          MOV     ADRF7+2,$REG1   ;
2639
2640  013070  012703  001170                  MOV     #$REG0,R3       ; ADDR(DATA)
2641  013074  172623                          LDF     (R3)+,AC2       ; M2-R3
2642
2643  013076  174237  001174                  STF     AC2,@#$REG2     ; M3-R7
2644
2645  013102  020327  001174                  CMP     R3,#$REG0+4     ; WAS FSRC ADDR REG INCRE RIGHT AMOUNT?
2646  013106  001401                          BEQ     66$             ;
2647  013110  104066                          ERROR   66              ; NOT EQUAL, SIGNAL ERROR
2648  013112                          66$:
2649
2650  013112  026767  166052  166054          CMP     $REG0,$REG2     ; 1ST WORD PATTERN CHECK?
2651  013120  001004                          BNE     64$             ;
2652  013122  026767  166044  166046          CMP     $REG1,$REG3     ; 2ND WORD PATTERN CHECK?
2653  013130  001401                          BEQ     65$             ;
2654  013132  104064                  64$:    ERROR   64              ; PATTERN DOESNT MATCH
2655
2656  013134                          65$:
2657  013134  000402                          BR      TST153          ;;
2658
2659  013136  170360  170360          ADRF7:  .WORD   ALT4N,ALT4N     ; TEST PATTERN
2660
2661
2662                                  ;;*****************************************************************
2663                                  ;*TEST 153       TEST OF LOAD-FSRC MODE-4, STORE-FDST MODE-5, F MODE
2664                                  ;;*****************************************************************
2665  013142  000004                  TST153: SCOPE
2666  013144  170127  047400                  LDFPS   #047400         ; F MODE FPS
2667  013150  016767  000076  166012          MOV     ADRF10+0,$REG0  ; MOVE PATTERN
2668  013156  016767  000072  166006          MOV     ADRF10+2,$REG1  ;
2669
2670  013164  012705  001174                  MOV     #$REG0+4,R5     ; ADDR(DATA+4)
```

```
2671  013170  172745                          LDF     -(R5),AC3       ; M4-R5
2672
2673  013172  012767  001174  166000          MOV     #$REG2,$REG4    ; ADDR(DEST)
2674  013200  012704  001202                  MOV     #$REG4+2,R4     ; ADDR(ADDR(DEST)+2)
2675  013204  174354                          STF     AC3,@-(R4)      ; M5-R4
2676
2677  013206  020527  001170                  CMP     R5,#$REG0       ; WAS FSRC ADDR REG DECRE RIGHT AMOUNT?
2678  013212  001401                          BEQ     66$             ;
2679  013214  104067                          ERROR   67              ; NOT EQUAL, SIGNAL ERROR
2680  013216                          66$:
2681
2682  013216  020427  001200                  CMP     R4,#$REG4       ; WAS FDST ADDR REG DECRE RIGHT AMOUNT?
2683  013222  001401                          BEQ     67$             ;
2684  013224  104070                          ERROR   70              ; NOT EQUAL, SIGNAL ERROR
2685  013226                          67$:
2686
2687  013226  026767  165736  165740          CMP     $REG0,$REG2     ; 1ST WORD PATTERN CHECK?
2688  013234  001004                          BNE     64$             ;
2689  013236  026767  165730  165732          CMP     $REG1,$REG3     ; 2ND WORD PATTERN CHECK?
2690  013244  001401                          BEQ     65$             ;
2691  013246  104071                  64$:    ERROR   71              ; PATTERN DOESNT MATCH
2692
2693  013250                          65$:
2694  013250  000402                          BR      TST154          ;;
2695
2696  013252  052525  052525          ADRF10: .WORD   ALTP,ALTP       ; TEST PATTERN
2697
2698
2699                                  ;;*****************************************************************
2700                                  ;;TEST 154      TEST OF LOAD-FSRC MODE-2(PC), STORE-FDST MODE-7(PC), F MODE
2701                                  ;;*****************************************************************
2702  013256  000004                  TST154: SCOPE
2703  013260  170127  047400                  LDFPS   #047400         ; F MODE FPS
2704  013264  016767  000070  165676          MOV     ADRF11+0,$REG0  ; MOVE PATTERN
2705  013272  016767  000064  165672          MOV     ADRF11+2,$REG1  ;
2706
2707  013300  016767  165664  000002          MOV     $REG0,1$        ; PUT DATA
2708  013306  172427                          LDF     (PC)+,AC0       ; M2-R7
2709  013310  000000                  1$:     .WORD   0               ; DATA
2710  013312  000403                          BR      2$              ; THIS SHOULD BE NEXT INSTR AFTER FPP PC MODE 2
2711  013314  104120                          ERROR   120             ; NOT HERE  (+4 INCRE)
2712  013316  104121                          ERROR   121             ; OR HERE  (+6 INCRE)
2713  013320  104122                          ERROR   122             ; OR HERE  (+10 INCRE)
2714
2715  013322  012767  001174  165650  2$:     MOV     #$REG2,$REG4    ; ADDR(DEST)
2716  013330  174077  165644                  STF     AC0,@$REG4      ; M7-R7
2717
2718  013334  026767  165630  165632          CMP     $REG0,$REG2     ; 1ST WORD PATTERN CHECK?
2719  013342  001004                          BNE     64$             ;
2720  013344  026767  165622  165624          CMP     $REG1,$REG3     ; 2ND WORD PATTERN CHECK?
2721  013352  001401                          BEQ     65$             ;
2722  013354  104072                  64$:    ERROR   72              ; PATTERN DOESNT MATCH
2723
2724  013356                          65$:
2725  013356  000402                          BR      TST155          ;;
2726
```

```
2727  013360  104117  000000          ADRF11: .WORD   ERROR117,0       ; TEST PATTERN
2728
2729
2730                                  ;;*****************************************************************
2731                                  ;*TEST 155      TEST OF LOAD-FSRC MODE-6(PC), STORE-FDST MODE-6(PC), F MODE
2732                                  ;;*****************************************************************
2733  013364  000004                  TST155: SCOPE
2734  013366  170127  047400                  LDFPS   #047400         ; F MODE FPS
2735  013372  016767  000044  165570          MOV     ADRF12+0,$REG0  ; MOVE PATTERN
2736  013400  016767  000040  165564          MOV     ADRF12+2,$REG1  ;
2737
2738  013406  172567  165556                  LDF     $REG0,AC1       ; M6-R7
2739
2740  013412  174167  165556                  STF     AC1,$REG2       ; M6-R7
2741
2742  013416  026767  165546  165550          CMP     $REG0,$REG2     ; 1ST WORD PATTERN CHECK?
2743  013424  001004                          BNE     64$             ;
2744  013426  026767  165540  165542          CMP     $REG1,$REG3     ; 2ND WORD PATTERN CHECK?
2745  013434  001401                          BEQ     65$             ;
2746  013436  104073                  64$:    ERROR   73              ; PATTERN DOESNT MATCH
2747
2748  013440                          65$:
2749  013440  000402                          BR      TST156          ;;
2750
2751  013442  052525  052525          ADRF12: .WORD   ALTP,ALTP       ; TEST PATTERN
2752
2753
2754                                  ;;*****************************************************************
2755                                  ;*TEST 156      TEST OF LOAD-FSRC MODE-0, STORE-FDST MODE-2(PC), F MODE
2756                                  ;;*****************************************************************
2757  013446  000004                  TST156: SCOPE
2758  013450  170127  047400                  LDFPS   #047400         ; F MODE FPS
2759  013454  016767  000062  165506          MOV     ADRF13+0,$REG0  ; MOVE PATTERN
2760  013462  016767  000056  165502          MOV     ADRF13+2,$REG1  ;
2761
2762  013470  172667  165474                  LDF     $REG0,AC2       ; M6-R7 WORKS
2763  013474  172702                          LDF     AC2,AC3         ; M0-R2
2764
2765  013476  012767  104117  000002          MOV     #ERROR117,1$    ; SETUP ERROR CALL FOR WRONG INCREMENT
2766  013504  174327                          STF     AC3,(PC)+       ; M2-R7
2767  013506  000000                  1$:     .WORD   0               ; DEST
2768  013510  000403                          BR      2$              ; THIS SHOULD BE NEXT INSTR AFTER FPP PC MODE 2
2769  013512  104120                          ERROR   120             ; NOT HERE  (+4 INCRE)
2770  013514  104121                          ERROR   121             ; OR HERE  (+6 INCRE)
2771  013516  104122                          ERROR   122             ; OR HERE  (+10 INCRE)
2772  013520  016767  177762  165446  2$:     MOV     1$,$REG2        ; GET DEST
2773
2774  013526  026767  165436  165440          CMP     $REG0,$REG2     ; WAS 1 WORD STORED OK?
2775  013534  001401                          BEQ     64$             ;
2776  013536  104073                          ERROR   73              ; NOT EQUAL, SIGNAL ERROR
2777  013540                          64$:
2778
2779  013540  000402                          BR      TST157          ;;
2780
2781  013542  007417  000000          ADRF13: .WORD   ALTMP,0         ; TEST PATTERN
2782
```

```
                                        ;;*****************************************************************
                                        ;*TEST 157      TEST OF LOAD-FSRC MODE-6, STORE-FDST MODE-0, F MODE
                                        ;;*****************************************************************
013546  000004                  TST157: SCOPE
013550  170127  047400                  LDFPS   #047400         ; F MODE FPS
013554  016767  000052  165406          MOV     ADRF14+0,$REG0  ; MOVE PATTERN
013562  016767  000046  165402          MOV     ADRF14+2,$REG1  ;

013570  012701  001200                  MOV     #$REG0+10,R1    ; ADDR(DATA)+10
013574  172461  177770                  LDF     -10(R1),AC0     ; M6-R1

013600  174001                          STF     AC0,AC1         ; M0-R1
013602  174167  165366                  STF     AC1,$REG2       ; M6-R7 WORKS

013606  026767  165356  165360          CMP     $REG0,$REG2     ; 1ST WORD PATTERN CHECK?
013614  001004                          BNE     64$             ;
013616  026767  165350  165352          CMP     $REG1,$REG3     ; 2ND WORD PATTERN CHECK?
013624  001401                          BEQ     65$             ;
013626  104074                  64$:    ERROR   74              ; PATTERN DOESNT MATCH

013630                          65$:
013630  000402                          BR      TST160          ;;

013632  007417  007417          ADRF14: .WORD   ALT4P,ALT4P     ; TEST PATTERN
```

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

                                ;;*****************************************************************
                                ;*TEST 160      TEST ALL FP-ACCUM ARE THERE
                                ;;*****************************************************************
013636  000004                  TST160: SCOPE
013640  170127  047400                  LDFPS   #047400         ; INITIAL FPS (SINGLE FLOAT, INTEGER)
013644  005037  001170                  CLR     $REG0           ; SET AC0 = 0,
013650  012737  000005  001172          MOV     #5,$REG1        ;     AC1 = 1,
013656  172437  001170                  LDF     $REG0,AC0       ;     AC2 = 2,
013662  174005                          STF     AC0,AC5         ;     AC3 = 3,
013664  005337  001172                  DEC     $REG1           ;     AC4 = 4,
013670  172437  001170                  LDF     $REG0,AC0       ;     AC5 = 5
013674  174004                          STF     AC0,AC4         ; ...
013676  005337  001172                  DEC     $REG1           ; ...
013702  172737  001170                  LDF     $REG0,AC3       ; ...
013706  005337  001172                  DEC     $REG1           ; ...
013712  172637  001170                  LDF     $REG0,AC2       ; ...
013716  005337  001172                  DEC     $REG1           ; ...
013722  172537  001170                  LDF     $REG0,AC1       ; ...
013726  005337  001172                  DEC     $REG1           ; ...
013732  172437  001170                  LDF     $REG0,AC0       ; ...
013736  005037  001170                  CLR     $REG0           ; NOW TEST THE ABOVE IS TRUE:
013742  005037  001172                  CLR     $REG1           ; ...
013746  174037  001174                  STF     AC0,$REG2       ; DOES AC0 = 0 ?
013752  005737  001174                  TST     $REG2           ; ...
013756  001004                          BNE     1$              ; ...
013760  023737  001172  001176          CMP     $REG1,$REG3     ; ...
013766  001401                          BEQ     11$             ; ...
013770  104075                  1$:     ERROR   75              ;   NO, ERROR
013772  005237  001172          11$:    INC     $REG1           ;   YES, INC FOR NEXT TEST
013776  174137  001174                  STF     AC1,$REG2       ; DOES AC1 = 1 ?
014002  005737  001174                  TST     $REG2           ; ...
014006  001004                          BNE     2$              ; ...
014010  023737  001172  001176          CMP     $REG1,$REG3     ; ...
014016  001401                          BEQ     21$             ; ...
014020  104075                  2$:     ERROR   75              ;   NO, ERROR
014022  005237  001172          21$:    INC     $REG1           ;   YES, INC FOR NEXT TEST
014026  174237  001174                  STF     AC2,$REG2       ; DOES AC2 = 2 ?
014032  005737  001174                  TST     $REG2           ; ...
014036  001004                          BNE     3$              ; ...
014040  023737  001172  001176          CMP     $REG1,$REG3     ; ...
014046  001401                          BEQ     31$             ; ...
014050  104075                  3$:     ERROR   75              ;   NO, ERROR
014052  005237  001172          31$:    INC     $REG1           ;   YES, INC FOR NEXT TEST
014056  174337  001174                  STF     AC3,$REG2       ; DOES AC3 = 3 ?
014062  005737  001174                  TST     $REG2           ; ...
014066  001004                          BNE     4$              ; ...
014070  023737  001172  001176          CMP     $REG1,$REG3     ; ...
014076  001401                          BEQ     41$             ; ...
014100  104075                  4$:     ERROR   75              ;   NO, ERROR
014102  005237  001172          41$:    INC     $REG1           ;   YES, INC FOR NEXT TEST
014106  172404                          LDF     AC4,AC0         ; DOES AC4 = 4 ?
```

```
2866  014110  174037  001174                    STF     AC0,$REG2       ; ...
2867  014114  005737  001174                    TST     $REG2           ; ...
2868  014120  001004                            BNE     5$              ; ...
2869  014122  023737  001172  001176            CMP     $REG1,$REG3     ; ...
2870  014130  001401                            BEQ     51$             ; ...
2871  014132  104075                    5$:     ERROR   75              ;   NO, ERROR
2872  014134  005237  001172            51$:    INC     $REG1           ;   YES, INC FOR NEXT TEST
2873  014140  172405                            LDF     AC5,AC0         ; DOES AC5 = 5 ?
2874  014142  174037  001174                    STF     AC0,$REG2       ; ...
2875  014146  005737  001174                    TST     $REG2           ; ...
2876  014152  001004                            BNE     6$              ; ...
2877  014154  023737  001172  001176            CMP     $REG1,$REG3     ; ...
2878  014162  001401                            BEQ     61$             ; ...
2879  014164  104075                    6$:     ERROR   75              ;   NO, ERROR
2880  014166                            61$:                            ;   YES, GO FOR NEXT TEST
```

```
                                ;;*****************************************************************
                                ;*TEST 161       TEST OF ABSF INSTR, DATA SET ABSF-1
                                ;*              TRUNCATE MODE, ALL INTERRUPT ENABLES ON
                                ;;*****************************************************************
014166  000004                  TST161: SCOPE
014170  012705  014202                  MOV     #ABSF1,R5       ; PTR TO TEST DATA SET
014174  004737  021704                  JSR     PC,@#ABSFT      ; GO TEST
                
014200  000407                          BR      TST162          ;;
                
014202                          ABSF1:  ; TEST DATA SET ABSF-1:
014202  000000  000000                  .WORD   0,0     ; INITIAL MEM FLOAT NUMBER
014206  000000  000000                  .WORD   0,0     ; EXPECTED FLOAT RESULT
014212  047453  047444                  .WORD   047453,047444   ; FPS:  BEFORE, AFTER
014216  000000                          .WORD   000000          ; FEC AFTER ( 0 = N/A )
                
                
                                ;;*****************************************************************
                                ;*TEST 162       TEST OF ABSF INSTR, DATA SET ABSF-2
                                ;*              TRUNCATE MODE, ALL INTERRUPT ENABLES ON
                                ;;*****************************************************************
014220  000004                  TST162: SCOPE
014222  012705  014234                  MOV     #ABSF2,R5       ; PTR TO TEST DATA SET
014226  004737  021704                  JSR     PC,@#ABSFT      ; GO TEST
                
014232  000407                          BR      TST163          ;;
                
014234                          ABSF2:  ; TEST DATA SET ABSF-2:
014234  052525  052525                  .WORD   ALTP,ALTP       ; INITIAL MEM FLOAT NUMBER
014240  052525  052525                  .WORD   ALTP,ALTP       ; EXPECTED FLOAT RESULT
014244  047557  047540                  .WORD   047557,047540   ; FPS:  BEFORE, AFTER
014250  000000                          .WORD   000000          ; FEC AFTER ( 0 = N/A )
                
                
                                ;;*****************************************************************
                                ;*TEST 163       TEST OF ABSF INSTR, DATA SET ABSF-3
                                ;*              ROUNDING MODE, ALL INTERRUPT ENABLES ON
                                ;;*****************************************************************
014252  000004                  TST163: SCOPE
014254  012705  014266                  MOV     #ABSF3,R5       ; PTR TO TEST DATA SET
014260  004737  021704                  JSR     PC,@#ABSFT      ; GO TEST
                
014264  000407                          BR      TST164          ;;
                
014266                          ABSF3:  ; TEST DATA SET ABSF-3:
014266  125252  125252                  .WORD   ALTN,ALTN       ; INITIAL MEM FLOAT NUMBER
014272  025252  125252                  .WORD   025252,ALTN     ; EXPECTED FLOAT RESULT
014276  047417  047400                  .WORD   047417,047400   ; FPS:  BEFORE, AFTER
014302  000000                          .WORD   000000          ; FEC AFTER ( 0 = N/A )
                
                
                                ;;*****************************************************************
                                ;*TEST 164       TEST OF ABSF INSTR, DATA SET ABSF-4
                                ;*              ROUNDING MODE, ALL INTERRUPT ENABLES ON
                                ;;*****************************************************************
014304  000004                  TST164: SCOPE
```

```
                        T164    TEST OF ABSF INSTR, DATA SET ABSF-4

2937  014306  012705  014320            MOV     #ABSF4,R5       ; PTR TO TEST DATA SET
2938  014312  004737  021704            JSR     PC,@#ABSFT      ; GO TEST
2939
2940  014316  000407                    BR      TST165          ;;
2941
2942  014320                    ABSF4:  ; TEST DATA SET ABSF-4:
2943  014320  177777  177777            .WORD   LGN,MI  ; INITIAL MEM FLOAT NUMBER
2944  014324  077777  177777            .WORD   LGP,MI  ; EXPECTED FLOAT RESULT
2945  014330  047517  047500            .WORD   047517,047500   ; FPS:  BEFORE, AFTER
2946  014334  000000                    .WORD   000000          ; FEC AFTER ( 0 = N/A )
2947
2948
2949                      ;;*****************************************************************
2950                      ;*TEST 165       TEST OF ABSF INSTR, DATA SET ABSF-5
2951                      ;*              TRUNCATE MODE, ALL INTERRUPT ENABLES ON
2952                      ;;*****************************************************************
2953  014336  000004      TST165: SCOPE
2954  014340  012705  014352            MOV     #ABSF5,R5       ; PTR TO TEST DATA SET
2955  014344  004737  021704            JSR     PC,@#ABSFT      ; GO TEST
2956
2957  014350  000407                    BR      TST166          ;;
2958
2959  014352                    ABSF5:  ; TEST DATA SET ABSF-5:
2960  014352  000200  000000            .WORD   SMP,0   ; INITIAL MEM FLOAT NUMBER
2961  014356  000200  000000            .WORD   SMP,0   ; EXPECTED FLOAT RESULT
2962  014362  047457  047440            .WORD   047457,047440   ; FPS:  BEFORE, AFTER
2963  014366  000000                    .WORD   000000          ; FEC AFTER ( 0 = N/A )
2964
2965
2966                      ;;*****************************************************************
2967                      ;*TEST 166       TEST OF ABSF INSTR, DATA SET ABSF-6
2968                      ;*              TRUNCATE MODE, ALL INTERRUPT ENABLES ON
2969                      ;;*****************************************************************
2970  014370  000004      TST166: SCOPE
2971  014372  012705  014404            MOV     #ABSF6,R5       ; PTR TO TEST DATA SET
2972  014376  004737  021704            JSR     PC,@#ABSFT      ; GO TEST
2973
2974  014402  000407                    BR      TST167          ;;
2975
2976  014404                    ABSF6:  ; TEST DATA SET ABSF-6:
2977  014404  077777  177777            .WORD   LGP,MI  ; INITIAL MEM FLOAT NUMBER
2978  014410  077777  177777            .WORD   LGP,MI  ; EXPECTED FLOAT RESULT
2979  014414  047557  047540            .WORD   047557,047540   ; FPS:  BEFORE, AFTER
2980  014420  000000                    .WORD   000000          ; FEC AFTER ( 0 = N/A )
2981
2982
2983                      ;;*****************************************************************
2984                      ;*TEST 167       TEST OF ABSF INSTR, DATA SET ABSF-7
2985                      ;*              ROUNDING MODE, ALL INTERRUPT ENABLES ON
2986                      ;;*****************************************************************
2987  014422  000004      TST167: SCOPE
2988  014424  012705  014436            MOV     #ABSF7,R5       ; PTR TO TEST DATA SET
2989  014430  004737  021704            JSR     PC,@#ABSFT      ; GO TEST
2990
2991  014434  000407                    BR      TST170          ;;
2992
```

```
                                T167    TEST OF ABSF INSTR, DATA SET ABSF-7

014436                          ABSF7:  ; TEST DATA SET ABSF-7:
014436  100200  000000                  .WORD   SMN,0   ; INITIAL MEM FLOAT NUMBER
014442  000200  000000                  .WORD   SMP,0   ; EXPECTED FLOAT RESULT
014446  047417  047400                  .WORD   047417,047400   ; FPS:  BEFORE, AFTER
014452  000000                          .WORD   000000          ; FEC AFTER ( 0 = N/A )


                                ;;*****************************************************************
                                ;*TEST 170       TEST OF ABSF INSTR, DATA SET ABSF-10
                                ;*              ROUNDING MODE, ALL INTERRUPT ENABLES ON
                                ;;*****************************************************************
014454  000004                  TST170: SCOPE
014456  012705  014470                  MOV     #ABSF10,R5      ; PTR TO TEST DATA SET
014462  004737  021704                  JSR     PC,@#ABSFT      ; GO TEST

014466  000407                          BR      TST171          ;;

014470                          ABSF10: ; TEST DATA SET ABSF-10:
014470  000177  177777                  .WORD   ZX1MP,M1        ; INITIAL MEM FLOAT NUMBER
014474  000000  000000                  .WORD   0,0     ; EXPECTED FLOAT RESULT
014500  047513  047504                  .WORD   047513,047504   ; FPS:  BEFORE, AFTER
014504  000000                          .WORD   000000          ; FEC AFTER ( 0 = N/A )


                                ;;*****************************************************************
                                ;*TEST 171       TEST OF ABSF INSTR, DATA SET ABSF-11
                                ;*              TRUNCATE MODE, ALL INTERRUPT ENABLES ON
                                ;;*****************************************************************
014506  000004                  TST171: SCOPE
014510  012705  014522                  MOV     #ABSF11,R5      ; PTR TO TEST DATA SET
014514  004737  021704                  JSR     PC,@#ABSFT      ; GO TEST

014520  000407                          BR      TST172          ;;

014522                          ABSF11: ; TEST DATA SET ABSF-11:
014522  100000  000000                  .WORD   M0,0    ; INITIAL MEM FLOAT NUMBER
014526  000000  000000                  .WORD   0,0     ; EXPECTED FLOAT RESULT
014532  047453  147444                  .WORD   047453,147444   ; FPS:  BEFORE, AFTER
014536  100014                          .WORD   100014          ; FEC AFTER ( 0 = N/A )


                                ;*****************************************************************
                                ;*TEST 172       TEST OF ABSF INSTR, DATA SET ABSF-12
                                ;*              TRUNCATE MODE, ALL INTERRUPT ENABLES ON
                                ;*****************************************************************
014540  000004                  TST172: SCOPE
014542  012705  014554                  MOV     #ABSF12,R5      ; PTR TO TEST DATA SET
014546  004737  021704                  JSR     PC,@#ABSFT      ; GO TEST

014552  000407                          BR      TST173          ;;

014554                          ABSF12: ; TEST DATA SET ABSF-12:
014554  100000  000001                  .WORD   M0,1    ; INITIAL MEM FLOAT NUMBER
014560  000000  000000                  .WORD   0,0     ; EXPECTED FLOAT RESULT
014564  047553  147544                  .WORD   047553,147544   ; FPS:  BEFORE, AFTER
014570  100014                          .WORD   100014          ; FEC AFTER ( 0 = N/A
```

```
3049
3050
3051                                    ;;*****************************************************************
3052                                    ;*TEST 173      TEST OF ABSF INSTR, DATA SET ABSF-13
3053                                    ;*             ROUNDING MODE, -0 INTERRUPT ENABLE OFF, ALL OTHERS ON
3054                                    ;;*****************************************************************
3055  014572  000004                    TST173: SCOPE
3056  014574  012705  014606                    MOV     #ABSF13,R5      ; PTR TO TEST DATA SET
3057  014600  004737  021704                    JSR     PC,@#ABSFT      ; GO TEST
3058
3059  014604  000407                            BR      TST174          ;;
3060
3061  014606                            ABSF13: ; TEST DATA SET ABSF-13:
3062  014606  100177  177777                    .WORD   ZX1MN,M1        ; INITIAL MEM FLOAT NUMBER
3063  014612  000000  000000                    .WORD   0,0     ; EXPECTED FLOAT RESULT
3064  014616  043413  043404                    .WORD   043413,043404   ; FPS:  BEFORE, AFTER
3065  014622  000000                            .WORD   000000          ; FEC AFTER ( 0 = N/A )
3066
3067
3068
```

```
3069                                    ;;*****************************************************************
3070                                    ;*TEST 174      TEST OF ABSD INSTR, DATA SET ABSD-1
3071                                    ;*             ROUNDING MODE, ALL INTERRUPT ENABLES ON
3072                                    ;;*****************************************************************
3073  014624  000004                    TST174: SCOPE
3074  014626  012705  014640                    MOV     #ABSD1,R5       ; PTR TO TEST DATA SET
3075  014632  004737  022060                    JSR     PC,@#ABSDT      ; GO TEST
3076
3077  014636  000413                            BR      TST175          ;;
3078
3079  014640                          ABSD1:    ; TEST DATA SET ABSD-1:
3080  014640  000000  000000  000000            .WORD   0,0,0,0 ; INITIAL MEM FLOAT NUMBER
3081  014646  000000
3082  014650  000000  000000  000000            .WORD   0,0,0,0 ; EXPECTED FLOAT RESULT
3083  014656  000000
3084  014660  047713  047704                    .WORD   047713,047704   ; FPS:  BEFORE, AFTER
3085  014664  000000                            .WORD   000000          ; FEC AFTER ( 0 = N/A )
3086
3087
3088                                    ;;*****************************************************************
3089                                    ;*TEST 175      TEST OF ABSD INSTR, DATA SET ABSD-2
3090                                    ;*             TRUNCATE MODE, ALL INTERRUPT ENABLES ON
3091                                    ;;*****************************************************************
3092  014666  000004                    TST175: SCOPE
3093  014670  012705  014702                    MOV     #ABSD2,R5       ; PTR TO TEST DATA SET
3094  014674  004737  022060                    JSR     PC,@#ABSDT      ; GO TEST
3095
3096  014700  000413                            BR      TST176          ;;
3097
3098  014702                          ABSD2:    ; TEST DATA SET ABSD-2:
3099  014702  052525  052525  052525            .WORD   ALTP,ALTP,ALTP,ALTP     ; INITIAL MEM FLOAT NUMBER
3100  014710  052525
3101  014712  052525  052525  052525            .WORD   ALTP,ALTP ALTP,ALTP     ; EXPECTED FLOAT RESULT
3102  014720  052525
3103  014722  047657  047640                    .WORD   047657,047640   ; FPS:  BEFORE, AFTER
3104  014726  000000                            .WORD   000000          ; FEC AFTER ( 0 = N/A )
3105
3106
3107                                    ;;*****************************************************************
3108                                    ;*TEST 176      TEST OF ABSD INSTR, DATA SET ABSD-3
3109                                    ;*             TRUNCATE MODE, ALL INTERRUPT ENABLES ON
3110                                    ;;*****************************************************************
3111  014730  000004                    TST176: SCOPE
3112  014732  012705  014744                    MOV     #ABSD3,R5       ; PTR TO TEST DATA SET
3113  014736  004737  022060                    JSR     PC,@#ABSDT      ; GO TEST
3114
3115  014742  000413                            BR      TST177          ;;
3116
3117  014744                          ABSD3:    ; TEST DATA SET ABSD-3:
3118  014744  125252  125252  125252            .WORD   ALTN,ALTN,ALTN,ALTN     ; INITIAL MEM FLOAT NUMBER
3119  014752  125252
3120  014754  025252  125252  125252            .WORD   025252,ALTN,ALTN,ALTN   ; EXPECTED FLOAT RESULT
3121  01[illegible]62  125252
3122  0[illegible]64  047757  047740            .WORD   047757,047740   ; FPS:  BEFORE, AFTER
3123  [illegible]70  000000                     .WORD   000000          ; FEC AFTER ( 0 = N/A )
3124
```

```
                                ;;*****************************************************************
                                ;*TEST 177       TEST OF ABSD INSTR, DATA SET ABSD-4
                                ;*              ROUNDING MODE, ALL INTERRUPT ENABLES ON
                                ;;*****************************************************************
014772  000004                  TST177: SCOPE
014774  012705  015006                  MOV     #ABSD4,R5       ; PTR TO TEST DATA SET
015000  004737  022060                  JSR     PC,@#ABSDT      ; GO TEST

015004  000413                          BR      TST200          ;;

015006                          ABSD4:  ; TEST DATA SET ABSD-4:
015006  177777  177777  177777          .WORD   LGN,M1,M1,M1    ; INITIAL MEM FLOAT NUMBER
015014  177777
015016  077777  177777  177777          .WORD   LGP,M1,M1,M1    ; EXPECTED FLOAT RESULT
015024  177777
015026  047617  047600                  .WORD   047617,047600   ; FPS:  BEFORE, AFTER
015032  000000                          .WORD   000000          ; FEC AFTER ( 0 = N/A )

                                ;;*****************************************************************
                                ;*TEST 200       TEST OF ABSD INSTR, DATA SET ABSD-5
                                ;*              ROUNDING MODE, ALL INTERRUPT ENABLES ON
                                ;;*****************************************************************
015034  000004                  TST200: SCOPE
015036  012705  015050                  MOV     #ABSD5,R5       ; PTR TO TEST DATA SET
015042  004737  022060                  JSR     PC,@#ABSDT      ; GO TEST

015046  000413                          BR      TST201          ;;

015050                          ABSD5:  ; TEST DATA SET ABSD-5:
015050  000200  000000  000000          .WORD   SMP,0,0,0       ; INITIAL MEM FLOAT NUMBER
015056  000000
015060  000200  000000  000000          .WORD   SMP,0,0,0       ; EXPECTED FLOAT RESULT
015066  000000
015070  047717  047700                  .WORD   047717,047700   ; FPS:  BEFORE, AFTER
015074  000000                          .WORD   000000          ; FEC AFTER ( 0 = N/A )

                                ;;*****************************************************************
                                ;*TEST 201       TEST OF ABSD INSTR, DATA SET ABSD-6
                                ;*              TRUNCATE MODE, ALL INTERRUPT ENABLES ON
                                ;;*****************************************************************
015076  000004                  TST201: SCOPE
015100  012705  015112                  MOV     #ABSD6,R5       ; PTR TO TEST DATA SET
015104  004737  022060                  JSR     PC,@#ABSDT      ; GO TEST

015110  000413                          BR      TST202          ;;

015112                          ABSD6:  ; TEST DATA SET ABSD-6:
015112  077777  177777  177777          .WORD   LGP,M1,M1,M1    ; INITIAL MEM FLOAT NUMBER
015120  177777
015122  077777  177777  177777          .WORD   LGP,M1,M1,M1    ; EXPECTED FLOAT RESULT
015130  177777
015132  047657  047640                  .WORD   047657,047640   ; FPS:  BEFORE, AFTER
015136  000000                          .WORD   000000          ; FEC AFTER ( 0 = N/A )
```

```
                                ;;*****************************************************************
                                ;*TEST 202       TEST OF ABSD INSTR, DATA SET ABSD-7
                                ;*              TRUNCATE MODE, ALL INTERRUPT ENABLES ON
                                ;;*****************************************************************
015140  000004                  TST202: SCOPE
015142  012705  015154                  MOV     #ABSD7,R5       ; PTR TO TEST DATA SET
015146  004737  022060                  JSR     PC,@#ABSDT      ; GO TEST
                                
015152  000413                          BR      TST203          ;;
                                
015154                          ABSD7:  ; TEST DATA SET ABSD-7:
015154  100200  000000  000000          .WORD   SMN,0,0,0       ; INITIAL MEM FLOAT NUMBER
015162  000000
015164  000200  000000  000000          .WORD   SMP,0,0,0       ; EXPECTED FLOAT RESULT
015172  000000
015174  047757  047740                  .WORD   047757,047740   ; FPS:  BEFORE, AFTER
015200  000000                          .WORD   000000          ; FEC AFTER ( 0 = N/A )
                                
                                
                                ;;*****************************************************************
                                ;*TEST 203       TEST OF ABSD INSTR, DATA SET ABSD-10
                                ;*              ROUNDING MODE, ALL INTERRUPT ENABLES ON
                                ;;*****************************************************************
015202  000004                  TST203: SCOPE
015204  012705  015216                  MOV     #ABSD10,R5      ; PTR TO TEST DATA SET
015210  004737  022060                  JSR     PC,@#ABSDT      ; GO TEST
                                
015214  000413                          BR      TST204          ;;
                                
015216                          ABSD10: ; TEST DATA SET ABSD-10:
015216  000177  177777  177777          .WORD   ZX1MP,M1,M1,M1  ; INITIAL MEM FLOAT NUMBER
015224  177777
015226  000000  000000  000000          .WORD   0,0,0,0 ; EXPECTED FLOAT RESULT
015234  000000
015236  047613  047604                  .WORD   047613,047604   ; FPS:  BEFORE, AFTER
015242  000000                          .WORD   000000          ; FEC AFTER ( 0 = N/A )
                                
                                
                                ;;*****************************************************************
                                ;*TEST 204       TEST OF ABSD INSTR, DATA SET ABSD-11
                                ;*              ROUNDING MODE, ALL INTERRUPT ENABLES ON
                                ;;*****************************************************************
015244  000004                  TST204: SCOPE
015246  012705  015260                  MOV     #ABSD11,R5      ; PTR TO TEST DATA SET
015252  004737  022060                  JSR     PC,@#ABSDT      ; GO TEST
                                
015256  000413                          BR      TST205          ;;
                                
015260                          ABSD11: ; TEST DATA SET ABSD-11:
015260  100000  000000  000000          .WORD   M0,0,0,0        ; INITIAL MEM FLOAT NUMBER
015266  000000
015270  000000  000000  000000          .WORD   0,0,0,0 ; EXPECTED FLOAT RESULT
015276  000000
015300  047713  147704                  .WORD   047713,147704   ; FPS:  BEFORE, AFTER
```

```
3237  015304  100014                          .WORD   100014          ; FEC AFTER ( 0 = N/A )
3238
3239
3240                                  ;;*****************************************************************
3241                                  ;*TEST 205       TEST OF ABSD INSTR, DATA SET ABSD-12
3242                                  ;*              TRUNCATE MODE, ALL INTERRUPT ENABLES ON
3243                                  ;;*****************************************************************
3244  015306  000004                  TST205: SCOPE
3245  015310  012705  015322                  MOV     #ABSD12,R5      ; PTR TO TEST DATA SET
3246  015314  004737  022060                  JSR     PC,@#ABSDT      ; GO TEST
3247
3248  015320  000413                          BR      TST206          ;;
3249
3250  015322                          ABSD12: ; TEST DATA SET ABSD-12:
3251  015322  100000  000000  000000          .WORD   M0,0,0,1        ; INITIAL MEM FLOAT NUMBER
3252  015330  000001
3253  015332  000000  000000  000000          .WORD   0,0,0,0 ; EXPECTED FLOAT RESULT
3254  015340  000000
3255  015342  047653  147644                  .WORD   047653,147644   ; FPS:  BEFORE, AFTER
3256  015346  100014                          .WORD   100014          ; FEC AFTER ( 0 = N/A )
3257
3258
3259                                  ;;*****************************************************************
3260                                  ;*TEST 206       TEST OF ABSD INSTR, DATA SET ABSD-13
3261                                  ;*              TRUNCATE MODE, -0 INTERRUPT ENABLE OFF, ALL OTHERS ON
3262                                  ;;*****************************************************************
3263  015350  000004                  TST206: SCOPE
3264  015352  012705  015364                  MOV     #ABSD13,R5      ; PTR TO TEST DATA SET
3265  015356  004737  022060                  JSR     PC,@#ABSDT      ; GO TEST
3266
3267  015362  000413                          BR      TST207          ;;
3268
3269  015364                          ABSD13: ; TEST DATA SET ABSD-13:
3270  015364  100177  177777  177777          .WORD   ZX1MN,M1,M1,M1  ; INITIAL MEM FLOAT NUMBER
3271  015372  177777
3272  015374  000000  000000  000000          .WORD   0,0,0,0 ; EXPECTED FLOAT RESULT
3273  015402  000000
3274  015404  043653  043644                  .WORD   043653,043644   ; FPS:  BEFORE, AFTER
3275  015410  000000                          .WORD   000000          ; FEC AFTER ( 0 = N/A )
3276
3277
3278
```

```
                                ;;*****************************************************************
                                ;*TEST 207       TEST OF NEGF INSTR, DATA SET NEGF-1
                                ;*              ROUNDING MODE, ALL INTERRUPT ENABLES ON
                                ;;*****************************************************************
015412  000004                  TST207: SCOPE
015414  012705  015426                  MOV     #NEGF1,R5       ; PTR TO TEST DATA SET
015420  004737  022270                  JSR     PC,@#NEGFT      ; GO TEST
                                
015424  000407                          BR      TST210          ;;
                                
015426                          NEGF1:  ; TEST DATA SET NEGF-1:
015426  000000  000000                  .WORD   0,0     ; INITIAL MEM FLOAT NUMBER
015432  000000  000000                  .WORD   0,0     ; EXPECTED FLOAT RESULT
015436  047413  047404                  .WORD   047413,047404   ; FPS:  BEFORE, AFTER
015442  000000                          .WORD   000000          ; FEC AFTER ( 0 = N/A )
                                
                                
                                ;;*****************************************************************
                                ;*TEST 210       TEST OF NEGF INSTR, DATA SET NEGF-2
                                ;*              TRUNCATE MODE, ALL INTERRUPT ENABLES ON
                                ;;*****************************************************************
015444  000004                  TST210: SCOPE
015446  012705  015460                  MOV     #NEGF2,R5       ; PTR TO TEST DATA SET
015452  004737  022270                  JSR     PC,@#NEGFT      ; GO TEST
                                
015456  000407                          BR      TST211          ;;
                                
015460                          NEGF2:  ; TEST DATA SET NEGF-2:
015460  052525  052525                  .WORD   ALTP,ALTP       ; INITIAL MEM FLOAT NUMBER
015464  152525  052525                  .WORD   152525,ALTP     ; EXPECTED FLOAT RESULT
015470  047547  047550                  .WORD   047547,047550   ; FPS:  BEFORE, AFTER
015474  000000                          .WORD   000000          ; FEC AFTER ( 0 = N/A )
                                
                                
                                ;;*****************************************************************
                                ;*TEST 211       TEST OF NEGF INSTR, DATA SET NEGF-3
                                ;*              ROUNDING MODE, ALL INTERRUPT ENABLES ON
                                ;;*****************************************************************
015476  000004                  TST211: SCOPE
015500  012705  015512                  MOV     #NEGF3,R5       ; PTR TO TEST DATA SET
015504  004737  022270                  JSR     PC,@#NEGFT      ; GO TEST
                                
015510  000407                          BR      TST212          ;;
                                
015512                          NEGF3:  ; TEST DATA SET NEGF-3:
015512  125252  125252                  .WORD   ALTN,ALTN       ; INITIAL MEM FLOAT NUMBER
015516  025252  125252                  .WORD   025252,ALTN     ; EXPECTED FLOAT RESULT
015522  047517  047500                  .WORD   047517,047500   ; FPS:  BEFORE, AFTER
015526  000000                          .WORD   000000          ; FEC AFTER ( 0 = N/A )
                                
                                
                                ;;*****************************************************************
                                ;*TEST 212       TEST OF NEGF INSTR, DATA SET NEGF-4
                                ;*              TRUNCATE MODE, ALL INTERRUPT ENABLES ON
                                ;;*****************************************************************
015530  000004                  TST212: SCOPE
```

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

                                        BR      TST213          ;;

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


                                ;;**********************************************************************
                                ;*TEST 213       TEST OF NEGF INSTR, DATA SET NEGF-5
                                ;*               ROUNDING MODE, ALL INTERRUPT ENABLES ON
                                ;;**********************************************************************
                                TST213: SCOPE
                                        MOV     #NEGF5,R5       ; PTR TO TEST DATA SET
                                        JSR     PC,@#NEGFT      ; GO TEST

                                        BR      TST214          ;;

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


                                ;;**********************************************************************
                                ;*TEST 214       TEST OF NEGF INSTR, DATA SET NEGF-6
                                ;*               TRUNCATE MODE, ALL INTERRUPT ENABLES ON
                                ;;**********************************************************************
                                TST214: SCOPE
                                        MOV     #NEGF6,R5       ; PTR TO TEST DATA SET
                                        JSR     PC,@#NEGFT      ; GO TEST

                                        BR      TST215          ;;

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


                                ;;**********************************************************************
                                ;*TEST 215       TEST OF NEGF INSTR, DATA SET NEGF-7
                                ;*               ROUNDING MODE, ALL INTERRUPT ENABLES ON
                                ;;**********************************************************************
                                TST215: SCOPE
                                        MOV     #NEGF7,R5       ; PTR TO TEST DATA SET
                                        JSR     PC,@#NEGFT      ; GO TEST

                                        BR      TST216          ;;
```

```
                                        T215    TEST OF NEGF INSTR, DATA SET NEGF-7

 3391  015662                           NEGF7:  ; TEST DATA SET NEGF-7:
 3392  015662  100200  000000                   .WORD   SMN,0   ; INITIAL MEM FLOAT NUMBER
 3393  015666  000200  000000                   .WORD   SMP,0   ; EXPECTED FLOAT RESULT
 3394  015672  047517  047500                   .WORD   047517,047500   ; FPS:  BEFORE, AFTER
 3395  015676  000000                           .WORD   000000          ; FEC AFTER ( 0 = N/A )
 3396
 3397
 3398                                   ;;*******************************************************************
 3399                                   ;*TEST 216      TEST OF NEGF INSTR, DATA SET NEGF-10
 3400                                   ;*             TRUNCATE MODE, ALL INTERRUPT ENABLES ON
 3401                                   ;;*******************************************************************
 3402  015700  000004                  TST216: SCOPE
 3403  015702  012705  015714                   MOV     #NEGF10,R5      ; PTR TO TEST DATA SET
 3404  015706  004737  022270                   JSR     PC,@#NEGFT      ; GO TEST
 3405
 3406  015712  000407                           BR      TST217          ;;
 3407
 3408  015714                           NEGF10: ; TEST DATA SET NEGF-10:
 3409  015714  000177  177777                   .WORD   ZX1MP,M1        ; INITIAL MEM FLOAT NUMBER
 3410  015720  000000  000000                   .WORD   0,0     ; EXPECTED FLOAT RESULT
 3411  015724  047453  047444                   .WORD   047453,047444   ; FPS:  BEFORE, AFTER
 3412  015730  000000                           .WORD   000000          ; FEC AFTER ( 0 = N/A )
 3413
 3414
 3415                                   ;;*******************************************************************
 3416                                   ;*TEST 217      TEST OF NEGF INSTR, DATA SET NEGF-11
 3417                                   ;*             TRUNCATE MODE, ALL INTERRUPT ENABLES ON
 3418                                   ;;*******************************************************************
 3419  015732  000004                  TST217: SCOPE
 3420  015734  012705  015746                   MOV     #NEGF11,R5      ; PTR TO TEST DATA SET
 3421  015740  004737  022270                   JSR     PC,@#NEGFT      ; GO TEST
 3422
 3423  015744  000407                           BR      TST220          ;;
 3424
 3425  015746                           NEGF11: ; TEST DATA SET NEGF-11:
 3426  015746  100000  000000                   .WORD   M0,0    ; INITIAL MEM FLOAT NUMBER
 3427  015752  000000  000000                   .WORD   0,0     ; EXPECTED FLOAT RESULT
 3428  015756  047453  147444                   .WORD   047453,147444   ; FPS:  BEFORE, AFTER
 3429  015762  100014                           .WORD   100014          ; FEC AFTER ( 0 = N/A )
 3430
 3431
 3432                                   ;;*******************************************************************
 3433                                   ;*TEST 220      TEST OF NEGF INSTR, DATA SET NEGF-12
 3434                                   ;*             TRUNCATE MODE, ALL INTERRUPT ENABLES ON
 3435                                   ;;*******************************************************************
 3436  015764  000004                  TST220: SCOPE
 3437  015766  012705  016000                   MOV     #NEGF12,R5      ; PTR TO TEST DATA SET
 3438  015772  004737  022270                   JSR     PC,@#NEGFT      ; GO TEST
 3439
 3440  015776  000407                           BR      TST221          ;;
 3441
 3442  016000                           NEGF12: ; TEST DATA SET NEGF-12:
 3443  016000  100000  000001                   .WORD   M0,1    ; INITIAL MEM FLOAT NUMBER
 3444  016004  000000  000000                   .WORD   0,0     ; EXPECTED FLOAT RESULT
 3445  016010  047553  147544                   .WORD   047553,147544   ; FPS:  BEFORE, AFTER
 3446  016014  100014                           .WORD   100014          ; FEC AFTER ( 0 = N/A )
```

```
3447
3448
3449                                    ;;*****************************************************************
3450                                    ;*TEST 221      TEST OF NEGF INSTR, DATA SET NEGF-13
3451                                    ;*             ROUNDING MODE, -0 INTERRUPT ENABLE OFF, ALL OTHERS ON
3452                                    ;;*****************************************************************
3453  016016  000004                    TST221: SCOPE
3454  016020  012705  016032                    MOV     #NEGF13,R5      ; PTR TO TEST DATA SET
3455  016024  004737  022270                    JSR     PC,@#NEGFT      ; GO TEST
3456
3457  016030  000407                            BR      TST222          ;;
3458
3459  016032                            NEGF13: ; TEST DATA SET NEGF-13:
3460  016032  100177  177777                    .WORD   2X1MN,M1        ; INITIAL MEM FLOAT NUMBER
3461  016036  000000  000000                    .WORD   0,0     ; EXPECTED FLOAT RESULT
3462  016042  043513  043504                    .WORD   043513,043504   ; FPS:  BEFORE, AFTER
3463  016046  000000                            .WORD   000000          ; FEC AFTER ( 0 = N/A )
3464
3465
3466
```

```
3467                                    ;;*****************************************************************
3468                                    ;*TEST 222      TEST OF NEGD INSTR, DATA SET NEGD-1
3469                                    ;*             TRUNCATE MODE, ALL INTERRUPT ENABLES ON
3470                                    ;;*****************************************************************
3471  016050  000004                    TST222: SCOPE
3472  016052  012705  016064                    MOV     #NEGD1,R5       ; PTR TO TEST DATA SET
3473  016056  004737  022444                    JSR     PC,@#NEGDT      ; GO TEST
3474
3475  016062  000413                            BR      TST223          ;;
3476
3477  016064                          NEGD1:  ; TEST DATA SET NEGD-1:
3478  016064  000000  000000  000000           .WORD   0,0,0,0 ; INITIAL MEM FLOAT NUMBER
3479  016072  000000
3480  016074  000000  000000  000000           .WORD   0,0,0,0 ; EXPECTED FLOAT RESULT
3481  016102  000000
3482  016104  047753  047744                   .WORD   047753,047744   ; FPS:  BEFORE, AFTER
3483  016110  000000                           .WORD   000000          ; FEC AFTER ( 0 = N/A )
3484
3485
3486                                    ;;*****************************************************************
3487                                    ;*TEST 223      TEST OF NEGD INSTR, DATA SET NEGD-2
3488                                    ;*             ROUNDING MODE, ALL INTERRUPT ENABLES ON
3489                                    ;;*****************************************************************
3490  016112  000004                    TST223: SCOPE
3491  016114  012705  016126                    MOV     #NEGD2,R5       ; PTR TO TEST DATA SET
3492  016120  004737  022444                    JSR     PC,@#NEGDT      ; GO TEST
3493
3494  016124  000413                            BR      TST224          ;;
3495
3496  016126                          NEGD2:  ; TEST DATA SET NEGD-2:
3497  016126  052525  052525  052525           .WORD   ALTP,ALTP,ALTP,ALTP     ; INITIAL MEM FLOAT NUMBER
3498  016134  052525
3499  016136  152525  052525  052525           .WORD   152525,ALTP,ALTP,ALTP   ; EXPECTED FLOAT RESULT
3500  016144  052525
3501  016146  047607  047610                   .WORD   047607,047610   ; FPS:  BEFORE, AFTER
3502  016152  000000                           .WORD   000000          ; FEC AFTER ( 0 = N/A )
3503
3504
3505                                    ;;*****************************************************************
3506                                    ;*TEST 224      TEST OF NEGD INSTR, DATA SET NEGD-3
3507                                    ;*             TRUNCATE MODE, ALL INTERRUPT ENABLES ON
3508                                    ;;*****************************************************************
3509  016154  000004                    TST224: SCOPE
3510  016156  012705  016170                    MOV     #NEGD3,R5       ; PTR TO TEST DATA SET
3511  016162  004737  022444                    JSR     PC,@#NEGDT      ; GO TEST
3512
3513  016166  000413                            BR      TST225          ;;
3514
3515  016170                          NEGD3:  ; TEST DATA SET NEGD-3:
3516  016170  125252  125252  125252           .WORD   ALTN,ALTN,ALTN,ALTN     ; INITIAL MEM FLOAT NUMBER
3517  016176  125252
3518  016200  025252  125252  125252           .WORD   025252,ALTN,ALTN,ALTN   ; EXPECTED FLOAT RESULT
3519  016206  125252
3520  016210  047657  047640                   .WORD   047657,047640   ; FPS:  BEFORE, AFTER
3521  016214  000000                           .WORD   000000          ; FEC AFTER ( 0 = N/A )
3522
```

```
                                ;;*****************************************************************
                                ;*TEST 225       TEST OF NEGD INSTR, DATA SET NEGD-4
                                ;*              ROUNDING MODE, ALL INTERRUPT ENABLES ON
                                ;;*****************************************************************
016216  000004                  TST225: SCOPE
016220  012705  016232                  MOV     #NEGD4,R5       ; PTR TO TEST DATA SET
016224  004737  022444                  JSR     PC,@#NEGDT      ; GO TEST
016230  000413                          BR      TST226          ;;
016232                          NEGD4:  ; TEST DATA SET NEGD-4:
016232  177777  177777  177777          .WORD   LGN,M1,M1,M1    ; INITIAL MEM FLOAT NUMBER
016240  177777
016242  077777  177777  177777          .WORD   LGP,M1,M1,M1    ; EXPECTED FLOAT RESULT
016250  177777
016252  047717  047700                  .WORD   047717,047700   ; FPS:  BEFORE, AFTER
016256  000000                          .WORD   000000          ; FEC AFTER ( 0 = N/A )

                                ;;*****************************************************************
                                ;*TEST 226       TEST OF NEGD INSTR, DATA SET NEGD-5
                                ;*              TRUNCATE MODE, ALL INTERRUPT ENABLES ON
                                ;;*****************************************************************
016260  000004                  TST226: SCOPE
016262  012705  016274                  MOV     #NEGD5,R5       ; PTR TO TEST DATA SET
016266  004737  022444                  JSR     PC,@#NEGDT      ; GO TEST
016272  000413                          BR      TST227          ;;
016274                          NEGD5:  ; TEST DATA SET NEGD-5:
016274  000200  000000  000000          .WORD   SMP,0,0,0       ; INITIAL MEM FLOAT NUMBER
016302  000000
016304  100200  000000  000000          .WORD   SMN,0,0,0       ; EXPECTED FLOAT RESULT
016312  000000
016314  047747  047750                  .WORD   047747,047750   ; FPS:  BEFORE, AFTER
016320  000000                          .WORD   000000          ; FEC AFTER ( 0 = N/A )

                                ;;*****************************************************************
                                ;*TEST 227       TEST OF NEGD INSTR, DATA SET NEGD-6
                                ;*              ROUNDING MODE, ALL INTERRUPT ENABLES ON
                                ;;*****************************************************************
016322  000004                  TST227: SCOPE
016324  012705  016336                  MOV     #NEGD6,R5       ; PTR TO TEST DATA SET
016330  004737  022444                  JSR     PC,@#NEGDT      ; GO TEST
016334  000413                          BR      TST230          ;;
016336                          NEGD6:  ; TEST DATA SET NEGD-6:
016336  077777  177777  177777          .WORD   LGP,M1,M1,M1    ; INITIAL MEM FLOAT NUMBER
016344  177777
016346  177777  177777  177777          .WORD   LGN,M1,M1,M1    ; EXPECTED FLOAT RESULT
016354  177777
016356  047607  047610                  .WORD   047607,047610   ; FPS:  BEFORE, AFTER
016362  000000                          .WORD   000000          ; FEC AFTER ( 0 = N/A )
```

```
                                ;;*****************************************************************
                                ;*TEST 230       TEST OF NEGD INSTR, DATA SET NEGD-7
                                ;*              TRUNCATE MODE, ALL INTERRUPT ENABLES ON
                                ;;*****************************************************************
016364  000004                  TST230: SCOPE
016366  012705  016400                  MOV     #NEGD7,R5       ; PTR TO TEST DATA SET
016372  004737  022444                  JSR     PC,@#NEGDT      ; GO TEST

016376  000413                          BR      TST231          ;;

016400                          NEGD7:  ; TEST DATA SET NEGD-7:
016400  100200  000000  000000          .WORD   SMN,0,0,0       ; INITIAL MEM FLOAT NUMBER
016406  000000
016410  000200  000000  000000          .WORD   SMP,0,0,0       ; EXPECTED FLOAT RESULT
016416  000000
016420  047657  047640                  .WORD   047657,047640   ; FPS:  BEFORE, AFTER
016424  000000                          .WORD   000000          ; FEC AFTER ( 0 = N/A )


                                ;;*****************************************************************
                                ;*TEST 231       TEST OF NEGD INSTR, DATA SET NEGD-10
                                ;*              ROUNDING MODE, ALL INTERRUPT ENABLES ON
                                ;;*****************************************************************
016426  000004                  TST231: SCOPE
016430  012705  016442                  MOV     #NEGD10,R5      ; PTR TO TEST DATA SET
016434  004737  022444                  JSR     PC,@#NEGDT      ; GO TEST

016440  000413                          BR      TST232          ;;

016442                          NEGD10: ; TEST DATA SET NEGD-10:
016442  000177  177777  177777          .WORD   ZX1MP,M1,M1,M1  ; INITIAL MEM FLOAT NUMBER
016450  177777
016452  000000  000000  000000          .WORD   0,0,0,0 ; EXPECTED FLOAT RESULT
016460  000000
016462  047713  047704                  .WORD   047713,047704   ; FPS:  BEFORE, AFTER
016466  000000                          .WORD   000000          ; FEC AFTER ( 0 = N/A )


                                ;;*****************************************************************
                                ;*TEST 232       TEST OF NEGD INSTR, DATA SET NEGD-11
                                ;*              ROUNDING MODE, ALL INTERRUPT ENABLES ON
                                ;;*****************************************************************
016470  000004                  TST232: SCOPE
016472  012705  016504                  MOV     #NEGD11,R5      ; PTR TO TEST DATA SET
016476  004737  022444                  JSR     PC,@#NEGDT      ; GO TEST

016502  000413                          BR      TST233          ;;

016504                          NEGD11: ; TEST DATA SET NEGD-11:
016504  100000  000000  000000          .WORD   M0,0,0,0        ; INITIAL MEM FLOAT NUMBER
016512  000000
016514  000000  000000  000000          .WORD   0,0,0,0 ; EXPECTED FLOAT RESULT
016522  000000
016524  047713  147704                  .WORD   047713,147704   ; FPS:  BEFORE, AFTER
```

```
3635  016530  100014                          .WORD   100014          ; FEC AFTER ( 0 = N/A )
3636
3637
3638                                  ;;*****************************************************************
3639                                  ;*TEST 233       TEST OF NEGD INSTR, DATA SET NEGD-12
3640                                  ;*              ROUNDING MODE, ALL INTERRUPT ENABLES ON
3641                                  ;;*****************************************************************
3642  016532  000004                  TST233: SCOPE
3643  016534  012705  016546                  MOV     #NEGD12,R5      ; PTR TO TEST DATA SET
3644  016540  004737  022444                  JSR     PC,@#NEGDT      ; GO TEST
3645
3646  016544  000413                          BR      TST234          ;;
3647
3648  016546                          NEGD12: ; TEST DATA SET NEGD-12:
3649  016546  100000  000000  000000          .WORD   M0,0,0,1        ; INITIAL MEM FLOAT NUMBER
3650  016554  000001
3651  016556  000000  000000  000000          .WORD   0,0,0,0 ; EXPECTED FLOAT RESULT
3652  016564  000000
3653  016566  047613  147604                  .WORD   047613,147604   ; FPS:  BEFORE, AFTER
3654  016572  100014                          .WORD   100014          ; FEC AFTER ( 0 = N/A )
3655
3656
3657                                  ;;*****************************************************************
3658                                  ;*TEST 234       TEST OF NEGD INSTR, DATA SET NEGD-13
3659                                  ;*              TRUNCATE MODE, -0 INTERRUPT ENABLE OFF, ALL OTHERS ON
3660                                  ;;*****************************************************************
3661  016574  000004                  TST234: SCOPE
3662  016576  012705  016610                  MOV     #NEGD13,R5      ; PTR TO TEST DATA SET
3663  016602  004737  022444                  JSR     PC,@#NEGDT      ; GO TEST
3664
3665  016606  000413                          BR      TST235          ;;
3666
3667  016610                          NEGD13: ; TEST DATA SET NEGD-13:
3668  016610  100177  177777  177777          .WORD   ZX1MN,M1,M1,M1  ; INITIAL MEM FLOAT NUMBER
3669  016616  177777
3670  016620  000000  000000  000000          .WORD   0,0,0,0 ; EXPECTED FLOAT RESULT
3671  016626  000000
3672  016630  043753  043744                  .WORD   043753,043744   ; FPS:  BEFORE, AFTER
3673  016634  000000                          .WORD   000000          ; FEC AFTER ( 0 = N/A )
3674
3675
3676
```

```
                                ;;*****************************************************************
                                ;*TEST 235       TEST OF CLRF INSTR, DATA SET CLRF-1
                                ;*              TRUNCATE MODE, ALL INTERRUPT ENABLES ON
                                ;;*****************************************************************
016636  000004                  TST235: SCOPE
016640  012705  016652                  MOV     #CLRF1,R5       ; PTR TO TEST DATA SET
016644  004737  022654                  JSR     PC,@#CLRFT      ; GO TEST
                                
016650  000407                          BR      TST236          ;;
                                
016652                          CLRF1:  ; TEST DATA SET CLRF-1:
016652  000000  000000                  .WORD   0,0     ; INITIAL MEM FLOAT NUMBER
016656  000000  000000                  .WORD   0,0     ; EXPECTED FLOAT RESULT
016662  047553  047544                  .WORD   047553,047544   ; FPS:  BEFORE, AFTER
016666  000000                          .WORD   000000          ; FEC AFTER ( 0 = N/A )
                                
                                
                                ;;*****************************************************************
                                ;*TEST 236       TEST OF CLRF INSTR, DATA SET CLRF-2
                                ;*              ROUNDING MODE, ALL INTERRUPT ENABLES ON
                                ;;*****************************************************************
016670  000004                  TST236: SCOPE
016672  012705  016704                  MOV     #CLRF2,R5       ; PTR TO TEST DATA SET
016676  004737  022654                  JSR     PC,@#CLRFT      ; GO TEST
                                
016702  000407                          BR      TST237          ;;
                                
016704                          CLRF2:  ; TEST DATA SET CLRF-2:
016704  177777  177777                  .WORD   M1,M1   ; INITIAL MEM FLOAT NUMBER
016710  000000  000000                  .WORD   0,0     ; EXPECTED FLOAT RESULT
016714  047413  047404                  .WORD   047413,047404   ; FPS:  BEFORE, AFTER
016720  000000                          .WORD   000000          ; FEC AFTER ( 0 = N/A )
                                
                                
                                ;;*****************************************************************
                                ;*TEST 237       TEST OF CLRF INSTR, DATA SET CLRF-3
                                ;*              TRUNCATE MODE, ALL INTERRUPT ENABLES ON
                                ;;*****************************************************************
016722  000004                  TST237: SCOPE
016724  012705  016736                  MOV     #CLRF3,R5       ; PTR TO TEST DATA SET
016730  004737  022654                  JSR     PC,@#CLRFT      ; GO TEST
                                
016734  000407                          BR      TST240          ;;
                                
016736                          CLRF3:  ; TEST DATA SET CLRF-3:
016736  052525  052525                  .WORD   ALTP,ALTP       ; INITIAL MEM FLOAT NUMBER
016742  000000  000000                  .WORD   0,0     ; EXPECTED FLOAT RESULT
016746  047453  047444                  .WORD   047453,047444   ; FPS:  BEFORE, AFTER
016752  000000                          .WORD   000000          ; FEC AFTER ( 0 = N/A )
                                
                                
                                ;;*****************************************************************
                                ;*TEST 240       TEST OF CLRF INSTR, DATA SET CLRF-4
                                ;*              ROUNDING MODE, ALL INTERRUPT ENABLES ON
                                ;;*****************************************************************
016754  000004                  TST240: SCOPE
```

```
3733  016756  012705  016770          MOV     #CLRF4,R5       ; PTR TO TEST DATA SET
3734  016762  004737  022654          JSR     PC,@#CLRFT      ; GO TEST
3735
3736  016766  000407                  BR      TST241          ;;
3737
3738  016770                  CLRF4:  ; TEST DATA SET CLRF-4:
3739  016770  125252  125252          .WORD   ALTN,ALTN       ; INITIAL MEM FLOAT NUMBER
3740  016774  000000  000000          .WORD   0,0     ; EXPECTED FLOAT RESULT
3741  017000  047513  047504          .WORD   047513,047504   ; FPS:  BEFORE, AFTER
3742  017004  000000                  .WORD   000000          ; FEC AFTER ( 0 = N/A )
3743
3744
3745                          ;;**********************************************************************
3746                          ;*TEST 241      TEST OF CLRF INSTR, DATA SET CLRF-5
3747                          ;*              TRUNCATE MODE, ALL INTERRUPT ENABLES ON
3748                          ;;**********************************************************************
3749  017006  000004          TST241: SCOPE
3750  017010  012705  017022          MOV     #CLRF5,R5       ; PTR TO TEST DATA SET
3751  017014  004737  022654          JSR     PC,@#CLRFT      ; GO TEST
3752
3753  017020  000407                  BR      TST242          ;;
3754
3755  017022                  CLRF5:  ; TEST DATA SET CLRF-5:
3756  017022  100000  000000          .WORD   MO,0    ; INITIAL MEM FLOAT NUMBER
3757  017026  000000  000000          .WORD   0,0     ; EXPECTED FLOAT RESULT
3758  017032  047553  047544          .WORD   047553,047544   ; FPS:  BEFORE, AFTER
3759  017036  000000                  .WORD   000000          ; FEC AFTER ( 0 = N/A )
3760
3761
3762                          ;;**********************************************************************
3763                          ;*TEST 242      TEST OF CLRF INSTR, DATA SET CLRF-6
3764                          ;*              ROUNDING MODE, ALL INTERRUPT ENABLES ON
3765                          ;;**********************************************************************
3766  017040  000004          TST242: SCOPE
3767  017042  012705  017054          MOV     #CLRF6,R5       ; PTR TO TEST DATA SET
3768  017046  004737  022654          JSR     PC,@#CLRFT      ; GO TEST
3769
3770  017052  000407                  BR      TST243          ;;
3771
3772  017054                  CLRF6:  ; TEST DATA SET CLRF-6:
3773  017054  000177  177777          .WORD   ZX1MP,M1        ; INITIAL MEM FLOAT NUMBER
3774  017060  000000  000000          .WORD   0,0     ; EXPECTED FLOAT RESULT
3775  017064  047413  047404          .WORD   047413,047404   ; FPS:  BEFORE, AFTER
3776  017070  000000                  .WORD   000000          ; FEC AFTER ( 0 = N/A )
3777
3778
3779
```

```
3780                                    ;;*****************************************************************
3781                                    ;*TEST 243      TEST OF CLRD INSTR, DATA SET CLRD-1
3782                                    ;*             ROUNDING MODE, ALL INTERRUPT ENABLES ON
3783                                    ;;*****************************************************************
3784  017072  000004                    TST243: SCOPE
3785  017074  012705  017106                    MOV     #CLRD1,R5       ; PTR TO TEST DATA SET
3786  017100  004737  023030                    JSR     PC,@#CLRDT      ; GO TEST
3787
3788  017104  000413                            BR      TST244          ;;
3789
3790  017106                          CLRD1:    ; TEST DATA SET CLRD-1:
3791  017106  000000  000000  000000            .WORD   0,0,0,0 ; INITIAL MEM FLOAT NUMBER
3792  017114  000000
3793  017116  000000  000000  000000            .WORD   0,0,0,0 ; EXPECTED FLOAT RESULT
3794  017124  000000
3795  017126  047613  047604                    .WORD   047613,047604   ; FPS:  BEFORE, AFTER
3796  017132  000000                            .WORD   000000          ; FEC AFTER ( 0 = N/A )
3797
3798
3799                                    ;;*****************************************************************
3800                                    ;*TEST 244      TEST OF CLRD INSTR, DATA SET CLRD-2
3801                                    ;*             TRUNCATE MODE, ALL INTERRUPT ENABLES ON
3802                                    ;;*****************************************************************
3803  017134  000004                    TST244: SCOPE
3804  017136  012705  017150                    MOV     #CLRD2,R5       ; PTR TO TEST DATA SET
3805  017142  004737  023030                    JSR     PC,@#CLRDT      ; GO TEST
3806
3807  017146  000413                            BR      TST245          ;;
3808
3809  017150                          CLRD2:    ; TEST DATA SET CLRD-2:
3810  017150  177777  177777  177777            .WORD   M1,M1,M1,M1     ; INITIAL MEM FLOAT NUMBER
3811  017156  177777
3812  017160  000000  000000  000000            .WORD   0,0,0,0 ; EXPECTED FLOAT RESULT
3813  017166  000000
3814  017170  047753  047744                    .WORD   047753,047744   ; FPS:  BEFORE, AFTER
3815  017174  000000                            .WORD   000000          ; FEC AFTER ( 0 = N/A )
3816
3817
3818                                    ;;*****************************************************************
3819                                    ;*TEST 245      TEST OF CLRD INSTR, DATA SET CLRD-3
3820                                    ;*             TRUNCATE MODE, ALL INTERRUPT ENABLES ON
3821                                    ;;*****************************************************************
3822  017176  000004                    TST245: SCOPE
3823  017200  012705  017212                    MOV     #CLRD3,R5       ; PTR TO TEST DATA SET
3824  017204  004737  023030                    JSR     PC,@#CLRDT      ; GO TEST
3825
3826  017210  000413                            BR      TST246          ;;
3827
3828  017212                          CLRD3:    ; TEST DATA SET CLRD-3:
3829  017212  052525  052525  052525            .WORD   ALTP,ALTP,ALTP,ALTP     ; INITIAL MEM FLOAT NUMBER
3830  017220  052525
3831  017222  000000  000000  000000            .WORD   0,0,0,0 ; EXPECTED FLOAT RESULT
3832  017230  000000
3833  017232  047653  047644                    .WORD   047653,047644   ; FPS:  BEFORE, AFTER
3834  017236  000000                            .WORD   000000          ; FEC AFTER ( 0 = N/A )
3835
```

```
                                ;;*****************************************************************
                                ;*TEST 246       TEST OF CLRD INSTR, DATA SET CLRD-4
                                ;*              ROUNDING MODE, ALL INTERRUPT ENABLES ON
                                ;;*****************************************************************
017240  000004                  TST246: SCOPE
017242  012705  017254                  MOV     #CLRD4,R5       ; PTR TO TEST DATA SET
017246  004737  023030                  JSR     PC,@#CLRDT      ; GO TEST

017252  000413                          BR      TST247          ;;

017254                          CLRD4:  ; TEST DATA SET CLRD-4:
017254  125252  125252  125252          .WORD   ALTN,ALTN,ALTN,ALTN     ; INITIAL MEM FLOAT NUMBER
017262  125252
017264  000000  000000  000000          .WORD   0,0,0,0 ; EXPECTED FLOAT RESULT
017272  000000
017274  047713  047704                  .WORD   047713,047704   ; FPS:  BEFORE, AFTER
017300  000000                          .WORD   000000          ; FEC AFTER ( 0 = N/A )


                                ;;*****************************************************************
                                ;*TEST 247       TEST OF CLRD INSTR, DATA SET CLRD-5
                                ;*              ROUNDING MODE, ALL INTERRUPT ENABLES ON
                                ;;*****************************************************************
017302  000004                  TST247: SCOPE
017304  012705  017316                  MOV     #CLRD5,R5       ; PTR TO TEST DATA SET
017310  004737  023030                  JSR     PC,@#CLRDT      ; GO TEST

017314  000413                          BR      TST250          ;;

017316                          CLRD5:  ; TEST DATA SET CLRD-5:
017316  100000  000000  000000          .WORD   M0,0,0,0        ; INITIAL MEM FLOAT NUMBER
017324  000000
017326  000000  000000  000000          .WORD   0,0,0,0 ; EXPECTED FLOAT RESULT
017334  000000
017336  047613  047604                  .WORD   047613,047604   ; FPS:  BEFORE, AFTER
017342  000000                          .WORD   000000          ; FEC AFTER ( 0 = N/A )


                                ;;*****************************************************************
                                ;*TEST 250       TEST OF CLRD INSTR, DATA SET CLRD-6
                                ;*              TRUNCATE MODE, ALL INTERRUPT ENABLES ON
                                ;;*****************************************************************
017344  000004                  TST250: SCOPE
017346  012705  017360                  MOV     #CLRD6,R5       ; PTR TO TEST DATA SET
017352  004737  023030                  JSR     PC,@#CLRDT      ; GO TEST

017356  000413                          BR      TST251          ;;

017360                          CLRD6:  ; TEST DATA SET CLRD-6:
017360  000177  177777  177777          .WORD   ZX1MP,M1,M1,M1  ; INITIAL MEM FLOAT NUMBER
017366  177777
017370  000000  000000  000000          .WORD   0,0,0,0 ; EXPECTED FLOAT RESULT
017376  000000
017400  047753  047744                  .WORD   047753,047744   ; FPS:  BEFORE, AFTER
017404  000000                          .WORD   000000          ; FEC AFTER ( 0 = N/A )
```

```
3892
3893
3894
```

```
                                ;;*****************************************************************
                                ;*TEST 251       TEST OF TSTF INSTR, DATA SET TSTF-1
                                ;*              ROUNDING MODE, ALL INTERRUPT ENABLES ON
                                ;;*****************************************************************
017406  000004                  TST251: SCOPE
017410  012705  017422                  MOV     #TSTF1,R5       ; PTR TO TEST DATA SET
017414  004737  023240                  JSR     PC,@#TSTFT      ; GO TEST
                                
017420  000407                          BR      TST252          ;;
                                
017422                          TSTF1:  ; TEST DATA SET TSTF-1:
017422  000000  000000                  .WORD   0,0     ; INITIAL MEM FLOAT NUMBER
017426  000000  000000                  .WORD   0,0     ; EXPECTED FLOAT RESULT
017432  047513  047504                  .WORD   047513,047504   ; FPS:  BEFORE, AFTER
017436  000000                          .WORD   000000          ; FEC AFTER ( 0 = N/A )
                                
                                
                                ;;*****************************************************************
                                ;*TEST 252       TEST OF TSTF INSTR, DATA SET TSTF-2
                                ;*              TRUNCATE MODE, ALL INTERRUPT ENABLES ON
                                ;;*****************************************************************
017440  000004                  TST252: SCOPE
017442  012705  017454                  MOV     #TSTF2,R5       ; PTR TO TEST DATA SET
017446  004737  023240                  JSR     PC,@#TSTFT      ; GO TEST
                                
017452  000407                          BR      TST253          ;;
                                
017454                          TSTF2:  ; TEST DATA SET TSTF-2:
017454  052525  052525                  .WORD   ALTP,ALTP       ; INITIAL MEM FLOAT NUMBER
017460  052525  052525                  .WORD   ALTP,ALTP       ; EXPECTED FLOAT RESULT
017464  047557  047540                  .WORD   047557,047540   ; FPS:  BEFORE, AFTER
017470  000000                          .WORD   000000          ; FEC AFTER ( 0 = N/A )
                                
                                
                                ;;*****************************************************************
                                ;*TEST 253       TEST OF TSTF INSTR, DATA SET TSTF-3
                                ;*              ROUNDING MODE, ALL INTERRUPT ENABLES ON
                                ;;*****************************************************************
017472  000004                  TST253: SCOPE
017474  012705  017506                  MOV     #TSTF3,R5       ; PTR TO TEST DATA SET
017500  004737  023240                  JSR     PC,@#TSTFT      ; GO TEST
                                
017504  000407                          BR      TST254          ;;
                                
017506                          TSTF3:  ; TEST DATA SET TSTF-3:
017506  125252  125252                  .WORD   ALTN,ALTN       ; INITIAL MEM FLOAT NUMBER
017512  125252  125252                  .WORD   ALTN,ALTN       ; EXPECTED FLOAT RESULT
017516  047407  047410                  .WORD   047407,047410   ; FPS:  BEFORE, AFTER
017522  000000                          .WORD   000000          ; FEC AFTER ( 0 = N/A )
                                
                                
                                ;;*****************************************************************
                                ;*TEST 254       TEST OF TSTF INSTR, DATA SET TSTF-4
                                ;*              TRUNCATE MODE, ALL INTERRUPT ENABLES ON
                                ;;*****************************************************************
017524  000004                  TST254: SCOPE
```

```
                                T254    TEST OF TSTF INSTR, DATA SET TSTF-4

017526  012705  017540                        MOV     #TSTF4,R5       ; PTR TO TEST DATA SET
017532  004737  023240                        JSR     PC,@#TSTFT      ; GO TEST

017536  000407                              BR      TST255          ;;

017540                          TSTF4:  ; TEST DATA SET TSTF-4:
017540  177777  177777                      .WORD   LQN,MI  ; INITIAL MEM FLOAT NUMBER
017544  177777  177777                      .WORD   LQN,MI  ; EXPECTED FLOAT RESULT
017550  047447  047450                      .WORD   047447,047450   ; FPS:  BEFORE, AFTER
017554  000000                              .WORD   000000          ; FEC AFTER ( 0 = N/A )


                                ;;**********************************************************************
                                ;*TEST 255       TEST OF TSTF INSTR, DATA SET TSTF-5
                                ;*               ROUNDING MODE, ALL INTERRUPT ENABLES ON
                                ;;**********************************************************************
017556  000004                  TST255: SCOPE
017560  012705  017572                    MOV     #TSTF5,R5       ; PTR TO TEST DATA SET
017564  004737  023240                    JSR     PC,@#TSTFT      ; GO TEST

017570  000407                          BR      TST256          ;;

017572                          TSTF5:  ; TEST DATA SET TSTF-5:
017572  000200  000000                  .WORD   SMP,0   ; INITIAL MEM FLOAT NUMBER
017576  000200  000000                  .WORD   SMP,0   ; EXPECTED FLOAT RESULT
017602  047517  047500                  .WORD   047517,047500   ; FPS:  BEFORE, AFTER
017606  000000                          .WORD   000000          ; FEC AFTER ( 0 = N/A )


                                ;;**********************************************************************
                                ;*TEST 256       TEST OF TSTF INSTR, DATA SET TSTF-6
                                ;*               TRUNCATE MODE, ALL INTERRUPT ENABLES ON
                                ;;**********************************************************************
017610  000004                  TST256: SCOPE
017612  012705  017624                    MOV     #TSTF6,R5       ; PTR TO TEST DATA SET
017616  004737  023240                    JSR     PC,@#TSTFT      ; GO TEST

017622  000407                          BR      TST257          ;;

017624                          TSTF6:  ; TEST DATA SET TSTF-6:
017624  077777  177777                  .WORD   LQP,MI  ; INITIAL MEM FLOAT NUMBER
017630  077777  177777                  .WORD   LQP,MI  ; EXPECTED FLOAT RESULT
017634  047557  047540                  .WORD   047557,047540   ; FPS:  BEFORE, AFTER
017640  000000                          .WORD   000000          ; FEC AFTER ( 0 = N/A )


                                ;;**********************************************************************
                                ;*TEST 257       TEST OF TSTF INSTR, DATA SET TSTF-7
                                ;*               ROUNDING MODE, ALL INTERRUPT ENABLES ON
                                ;;**********************************************************************
017642  000004                  TST257: SCOPE
017644  012705  017656                    MOV     #TSTF7,R5       ; PTR TO TEST DATA SET
017650  004737  023240                    JSR     PC,@#TSTFT      ; GO TEST

017654  000407                          BR      TST260          ;;
```

```
                                TSTF7:  ; TEST DATA SET TSTF-7:
017656                                  .WORD   SMN,0   ; INITIAL MEM FLOAT NUMBER
017656  100200  000000                  .WORD   SMN,0   ; EXPECTED FLOAT RESULT
017662  100200  000000                  .WORD   047407,047410   ; FPS:  BEFORE, AFTER
017666  047407  047410                  .WORD   000000          ; FEC AFTER ( 0 = N/A )
017672  000000
                                ;;**********************************************************************
                                ;*TEST 260       TEST OF TSTF INSTR, DATA SET TSTF-10
                                ;*               TRUNCATE MODE, ALL INTERRUPT ENABLES ON
                                ;;**********************************************************************
017674  000004                  TST260: SCOPE
017676  012705  017710                  MOV     #TSTF10,R5      ; PTR TO TEST DATA SET
017702  004737  023240                  JSR     PC,@#TSTFT      ; GO TEST
017706  000407                          BR      TST261          ;;
017710                          TSTF10: ; TEST DATA SET TSTF-10:
017710  000177  177777                  .WORD   ZX1MP,M1        ; INITIAL MEM FLOAT NUMBER
017714  000177  177777                  .WORD   ZX1MP,M1        ; EXPECTED FLOAT RESULT
017720  047453  047444                  .WORD   047453,047444   ; FPS:  BEFORE, AFTER
017724  000000                          .WORD   000000          ; FEC AFTER ( 0 = N/A )
                                ;;**********************************************************************
                                ;*TEST 261       TEST OF TSTF INSTR, DATA SET TSTF-11
                                ;*               ROUNDING MODE, ALL INTERRUPT ENABLES ON
                                ;;**********************************************************************
017726  000004                  TST261: SCOPE
017730  012705  017742                  MOV     #TSTF11,R5      ; PTR TO TEST DATA SET
017734  004737  023240                  JSR     PC,@#TSTFT      ; GO TEST
017740  000407                          BR      TST262          ;;
017742                          TSTF11: ; TEST DATA SET TSTF-11:
017742  100000  000000                  .WORD   M0,0    ; INITIAL MEM FLOAT NUMBER
017746  100000  000000                  .WORD   M0,0    ; EXPECTED FLOAT RESULT
017752  047503  147514                  .WORD   047503,147514   ; FPS:  BEFORE, AFTER
017756  100014                          .WORD   100014          ; FEC AFTER ( 0 = N/A )
                                ;;**********************************************************************
                                ;*TEST 262       TEST OF TSTF INSTR, DATA SET TSTF-12
                                ;*               TRUNCATE MODE, ALL INTERRUPT ENABLES ON
                                ;;**********************************************************************
017760  000004                  TST262: SCOPE
017762  012705  017774                  MOV     #TSTF12,R5      ; PTR TO TEST DATA SET
017766  004737  023240                  JSR     PC,@#TSTFT      ; GO TEST
017772  000407                          BR      TST263          ;;
017774                          TSTF12: ; TEST DATA SET TSTF-12:
017774  100000  000001                  .WORD   M0,1    ; INITIAL MEM FLOAT NUMBER
020000  100000  000001                  .WORD   M0,1    ; EXPECTED FLOAT RESULT
020004  047543  147554                  .WORD   047543,147554   ; FPS:  BEFORE, AFTER
020010  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
                                        ;;*****************************************************************
020012  000004                          TST263: SCOPE
020014  012705  020026                          MOV     #TSTF13,R5      ; PTR TO TEST DATA SET
020020  004737  023240                          JSR     PC,@#TSTFT      ; GO TEST
020024  000407                                  BR      TST264          ;;
020026                                  TSTF13: ; TEST DATA SET TSTF-13:
020026  100177  177777                          .WORD   ZX1MN,M1        ; INITIAL MEM FLOAT NUMBER
020032  100177  177777                          .WORD   ZX1MN,M1        ; EXPECTED FLOAT RESULT
020036  043403  043414                          .WORD   043403,043414   ; FPS:  BEFORE, AFTER
020042  000000                                  .WORD   000000          ; FEC AFTER ( 0 = N/A )
```

```
                                ;;*****************************************************************
                                ;*TEST 264       TEST OF TSTD INSTR, DATA SET TSTD-1
                                ;*              TRUNCATE MODE, ALL INTERRUPT ENABLES ON
                                ;;*****************************************************************
020044  000004                  TST264: SCOPE
020046  012705  020060                  MOV     #TSTD1,R5       ; PTR TO TEST DATA SET
020052  004737  023414                  JSR     PC,@#TSTDT      ; GO TEST
020056  000413                          BR      TST265          ;;
020060                          TSTD1:  ; TEST DATA SET TSTD-1:
020060  000000  000000  000000          .WORD   0,0,0,0 ; INITIAL MEM FLOAT NUMBER
020066  000000
020070  000000  000000  000000          .WORD   0,0,0,0 ; EXPECTED FLOAT RESULT
020076  000000
020100  047653  047644                  .WORD   047653,047644   ; FPS:  BEFORE, AFTER
020104  000000                          .WORD   000000          ; FEC AFTER ( 0 = N/A )


                                ;;*****************************************************************
                                ;*TEST 265       TEST OF TSTD INSTR, DATA SET TSTD-2
                                ;*              ROUNDING MODE, ALL INTERRUPT ENABLES ON
                                ;;*****************************************************************
020106  000004                  TST265: SCOPE
020110  012705  020122                  MOV     #TSTD2,R5       ; PTR TO TEST DATA SET
020114  004737  023414                  JSR     PC,@#TSTDT      ; GO TEST
020120  000413                          BR      TST266          ;;
020122                          TSTD2:  ; TEST DATA SET TSTD-2:
020122  052525  052525  052525          .WORD   ALTP,ALTP,ALTP,ALTP     ; INITIAL MEM FLOAT NUMBER
020130  052525
020132  052525  052525  052525          .WORD   ALTP,ALTP,ALTP,ALTP     ; EXPECTED FLOAT RESULT
020140  052525
020142  047717  047700                  .WORD   047717,047700   ; FPS:  BEFORE, AFTER
020146  000000                          .WORD   000000          ; FEC AFTER ( 0 = N/A )


                                ;;*****************************************************************
                                ;*TEST 266       TEST OF TSTD INSTR, DATA SET TSTD-3
                                ;*              TRUNCATE MODE, ALL INTERRUPT ENABLES ON
                                ;;*****************************************************************
020150  000004                  TST266: SCOPE
020152  012705  020164                  MOV     #TSTD3,R5       ; PTR TO TEST DATA SET
020156  004737  023414                  JSR     PC,@#TSTDT      ; GO TEST
020162  000413                          BR      TST267          ;;
020164                          TSTD3:  ; TEST DATA SET TSTD-3:
020164  125252  125252  125252          .WORD   ALTN,ALTN,ALTN,ALTN     ; INITIAL MEM FLOAT NUMBER
020172  125252
020174  125252  125252  125252          .WORD   ALTN,ALTN,ALTN,ALTN     ; EXPECTED FLOAT RESULT
020202  125252
020204  047747  047750                  .WORD   047747,047750   ; FPS:  BEFORE, AFTER
020210  000000                          .WORD   000000          ; FEC AFTER ( 0 = N/A )
```

```
                                ;;*****************************************************************
                                ;*TEST 267       TEST OF TSTD INSTR, DATA SET TSTD-4
                                ;*              ROUNDING MODE, ALL INTERRUPT ENABLES ON
                                ;;*****************************************************************
020212  000004                  TST267: SCOPE
020214  012705  020226                  MOV     #TSTD4,R5       ; PTR TO TEST DATA SET
020220  004737  023414                  JSR     PC,@#TSTDT      ; GO TEST

020224  000413                          BR      TST270          ;;

020226                          TSTD4:  ; TEST DATA SET TSTD-4:
020226  177777  177777  177777          .WORD   LGN,M1,M1,M1    ; INITIAL MEM FLOAT NUMBER
020234  177777
020236  177777  177777  177777          .WORD   LGN,M1,M1,M1    ; EXPECTED FLOAT RESULT
020244  177777
020246  047607  047610                  .WORD   047607,047610   ; FPS:  BEFORE, AFTER
020252  000000                          .WORD   000000          ; FEC AFTER ( 0 = N/A )


                                ;;*****************************************************************
                                ;*TEST 270       TEST OF TSTD INSTR, DATA SET TSTD-5
                                ;*              TRUNCATE MODE, ALL INTERRUPT ENABLES ON
                                ;;*****************************************************************
020254  000004                  TST270: SCOPE
020256  012705  020270                  MOV     #TSTD5,R5       ; PTR TO TEST DATA SET
020262  004737  023414                  JSR     PC,@#TSTDT      ; GO TEST

020266  000413                          BR      TST271          ;;

020270                          TSTD5:  ; TEST DATA SET TSTD-5:
020270  000200  000000  000000          .WORD   SMP,0,0,0       ; INITIAL MEM FLOAT NUMBER
020276  000000
020300  000200  000000  000000          .WORD   SMP,0,0,0       ; EXPECTED FLOAT RESULT
020306  000000
020310  047657  047640                  .WORD   047657,047640   ; FPS:  BEFORE, AFTER
020314  000000                          .WORD   000000          ; FEC AFTER ( 0 = N/A )


                                ;;*****************************************************************
                                ;*TEST 271       TEST OF TSTD INSTR, DATA SET TSTD-6
                                ;*              ROUNDING MODE, ALL INTERRUPT ENABLES ON
                                ;;*****************************************************************
020316  000004                  TST271: SCOPE
020320  012705  020332                  MOV     #TSTD6,R5       ; PTR TO TEST DATA SET
020324  004737  023414                  JSR     PC,@#TSTDT      ; GO TEST

020330  000413                          BR      TST272          ;;

020332                          TSTD6:  ; TEST DATA SET TSTD-6:
020332  077777  177777  177777          .WORD   LGP,M1,M1,M1    ; INITIAL MEM FLOAT NUMBER
020340  177777
020342  077777  177777  177777          .WORD   LGP,M1,M1,M1    ; EXPECTED FLOAT RESULT
020350  177777
020352  047717  047700                  .WORD   047717,047700   ; FPS:  BEFORE, AFTER
020356  000000                          .WORD   000000          ; FEC AFTER ( 0 = N/A )
```

```
                                ;;*****************************************************************
                                ;*TEST 272       TEST OF TSTD INSTR, DATA SET TSTD-7
                                ;*              TRUNCATE MODE, ALL INTERRUPT ENABLES ON
                                ;;*****************************************************************
020360  000004                  TST272: SCOPE
020362  012705  020374                  MOV     #TSTD7,R5       ; PTR TO TEST DATA SET
020366  004737  023414                  JSR     PC,@#TSTDT      ; GO TEST

020372  000413                          BR      TST273          ;;

020374                          TSTD7:  ; TEST DATA SET TSTD-7:
020374  100200  000000  000000          .WORD   SMN,0,0,0       ; INITIAL MEM FLOAT NUMBER
020402  000000
020404  100200  000000  000000          .WORD   SMN,0,0,0       ; EXPECTED FLOAT RESULT
020412  000000
020414  047747  047750                  .WORD   047747,047750   ; FPS:  BEFORE, AFTER
020420  000000                          .WORD   000000          ; FEC AFTER ( 0 = N/A )


                                ;;*****************************************************************
                                ;*TEST 273       TEST OF TSTD INSTR, DATA SET TSTD-10
                                ;*              ROUNDING MODE, ALL INTERRUPT ENABLES ON
                                ;;*****************************************************************
020422  000004                  TST273: SCOPE
020424  012705  020436                  MOV     #TSTD10,R5      ; PTR TO TEST DATA SET
020430  004737  023414                  JSR     PC,@#TSTDT      ; GO TEST

020434  000413                          BR      TST274          ;;

020436                          TSTD10: ; TEST DATA SET TSTD-10:
020436  000177  177777  177777          .WORD   ZX1MP,M1,M1,M1  ; INITIAL MEM FLOAT NUMBER
020444  177777
020446  000177  177777  177777          .WORD   ZX1MP,M1,M1,M1  ; EXPECTED FLOAT RESULT
020454  177777
020456  047613  047604                  .WORD   047613,047604   ; FPS:  BEFORE, AFTER
020462  000000                          .WORD   000000          ; FEC AFTER ( 0 = N/A )


                                ;;*****************************************************************
                                ;*TEST 274       TEST OF TSTD INSTR, DATA SET TSTD-11
                                ;*              TRUNCATE MODE, ALL INTERRUPT ENABLES ON
                                ;;*****************************************************************
020464  000004                  TST274: SCOPE
020466  012705  020500                  MOV     #TSTD11,R5      ; PTR TO TEST DATA SET
020472  004737  023414                  JSR     PC,@#TSTDT      ; GO TEST

020476  000413                          BR      TST275          ;;

020500                          TSTD11: ; TEST DATA SET TSTD-11:
020500  100000  000000  000000          .WORD   M0,0,0,0        ; INITIAL MEM FLOAT NUMBER
020506  000000
020510  100000  000000  000000          .WORD   M0,0,0,0        ; EXPECTED FLOAT RESULT
020516  000000
020520  047643  147654                  .WORD   047643,147654   ; FPS:  BEFORE, AFTER
```

```
004251  020524  100014                          .WORD   100014          ; FEC AFTER ( 0 = N/A )
004252
004253
004254                                  ;;*****************************************************************
004255                                  ;*TEST 275       TEST OF TSTD INSTR, DATA SET TSTD-12
004256                                  ;*              ROUNDING MODE, ALL INTERRUPT ENABLES ON
004257                                  ;;*****************************************************************
004258  020526  000004                  TST275: SCOPE
004259  020530  012705  020542                  MOV     #TSTD12,R5      ; PTR TO TEST DATA SET
004260  020534  004737  023414                  JSR     PC,@#TSTDT      ; GO TEST
004261
004262  020540  000413                          BR      TST276          ;;
004263
004264  020542                          TSTD12: ; TEST DATA SET TSTD-12:
004265  020542  100000  000000  000000          .WORD   M0,0,0,1        ; INITIAL MEM FLOAT NUMBER
004266  020550  000001
004267  020552  100000  000000  000000          .WORD   M0,0,0,1        ; EXPECTED FLOAT RESULT
004268  020560  000001
004269  020562  047703  147714                  .WORD   047703,147714   ; FPS:  BEFORE, AFTER
004270  020566  100014                          .WORD   100014          ; FEC AFTER ( 0 = N/A )
004271
004272
004273                                  ;;*****************************************************************
004274                                  ;*TEST 276       TEST OF TSTD INSTR, DATA SET TSTD-13
004275                                  ;*              TRUNCATE MODE, -0 INTERRUPT ENABLE OFF, ALL OTHERS ON
004276                                  ;;*****************************************************************
004277  020570  000004                  TST276: SCOPE
004278  020572  012705  020604                  MOV     #TSTD13,R5      ; PTR TO TEST DATA SET
004279  020576  004737  023414                  JSR     PC,@#TSTDT      ; GO TEST
004280
004281  020602  000413                          BR      TST277          ;;
004282
004283  020604                          TSTD13: ; TEST DATA SET TSTD-13:
004284  020604  100177  177777  177777          .WORD   ZX1MN,M1,M1,M1  ; INITIAL MEM FLOAT NUMBER
004285  020612  177777
004286  020614  100177  177777  177777          .WORD   ZX1MN,M1,M1,M1  ; EXPECTED FLOAT RESULT
004287  020622  177777
004288  020624  043643  043654                  .WORD   043643,043654   ; FPS:  BEFORE, AFTER
004289  020630  000000                          .WORD   000000          ; FEC AFTER ( 0 = N/A )
004290
004291
004292
```

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

020632                          TST277: ;FORCE LAST TEST NUMBER
020632  000004                          $COPE           ;CHECK FOR TEST ITERATIONS HERE
020634  005037  001310                  CLR     $TIMES          ;DONT ITERATE THIS "TEST"
020640  005037  001104                  CLR     $ERFLG          ;NO ERRORS HERE
020644  005037  001102                  CLR     $TSTNM          ;ZAP TEST ## WHEN DONE WITH A PASS

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

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

020650  076600  000022                  MED     ,RWHAMI         ;GET WHAMI INTO R0
020654  032700  000020                  BIT     #BIT04,R0       ;1=HFP PRESENT, 0=NONE
020660  001423                          BEQ     $EOP            ;EXIT IF NONE

020662  032777  000002  160254          BIT     #SW01,@SWR      ;1=HFP OR WFP TEST ONLY
020670  001017                          BNE     $EOP            ;0=ALTERNATE HFP AND WFP TESTS

020672  012701  010000                  MOV     #BIT12,R1       ;HFP PRESENT, AND IN ALTERNATE MODE;
020676  076600  000144                  MED     ,RFLAG          ;SO READ FLAGS,
020702  030100                          BIT     R1,R0           ;COMPLEMENT FLAG(5)=BIT12=HFP ENABLE FLAG
020704  001402                          BEQ     1$              ;
020706  040100                          BIC     R1,R0           ;CLEAR BIT 12
020710  000401                          BR      2$              ;
020712  050100                  1$:     BIS     R1,R0           ;SET BIT 12
020714  076600  000344          2$:     MED     ,WFLAG          ;REWRITE FLAGS

020720  030100                          BIT     R1,R0           ;HFP OR WFP NEXT ?
020722  001002                          BNE     $EOP            ;IF HFP AGAIN, START NEW PASS
020724  000137  003472                  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

020730                          $EOP:                           ;
020730  005037  001104                  CLR     $ERFLG          ;ZERO ERROR COUNT
020734  005037  001102                  CLR     $TSTNM          ;ZERO TEST NUMBER
020740  005037  001310                  CLR     $TIMES          ;ZERO NUMBER OF ITERATIONS
020744  005237  001332                  INC     $PASS           ;INCREMENT PASS COUNT,
020750  042737  100000  001332          BIC     #100000,$PASS   ;  BUT NEVER LET IN GO NEGATIVE
```

```
 4349  020756  005327                    DEC     (PC)+           ;PASS LOOP ?
 4350  020760  000001          $EOPCT:   .WORD   1               ;
 4351  020762  003021                    BGT     $DOAGN          ;YES
 4352  020764  012737                    MOV     (PC)+,@(PC)+    ;RESTORE COUNTER
 4353  020766  000001          $ENDCT:   .WORD   1               ;
 4354  020770  020760                    $EOPCT                  ;
 4355  020772  032777  002000  160144    BIT     #SW10,@SWR      ;BELL ON PASS END ?
 4356  021000  001002                    BNE     $GET42          ;NO
 4357  021002  104401  001314            TYPE    ,$BELL          ;YES
 4358
 4359  021006  013700  000042  $GET42:   MOV     @#42,R0         ;GET MONITOR ADDRESS
 4360  021012  001405                    BEQ     $DOAGN          ;NO MONITOR
 4361  021014  000005                    RESET                   ;CLEAR WORLD
 4362
 4363  021016  004710          $ENDAD:   JSR     PC,(R0)         ;GO TO MONITOR
 4364  021020  000240                    NOP                     ;
 4365  021022  000240                    NOP                     ;RESERVED FOR ACT11
 4366  021024  000240                    NOP                     ;
 4367
 4368  021026  000137  003434  $DOAGN:   JMP     @#NEWPAS        ;RETURN
 4369
 4370
```

```
                                        .SBTTL  SUBR TO PERFORM TEST OF LDD/STD
                                        
021032                                  LDADT:
021032  012700  000013                          MOV     #13,R0          ; LOAD $TMP0-12
021036  010501                                 MOV     R5,R1           ;   WITH TEST DATA SETS
021040  012702  001230                         MOV     #$TMP0,R2       ;   FOR DISPLAY LATER
021044  012122                                 MOV     (R1)+,(R2)+     ;
021046  077002                                 SOB     R0,.-2          ;
021050  012737  021056  001112                 MOV     #LDADL,$LPERR   ; ERROR LOOPING ADDRESS
                                        
021056  170165  000020                  LDADL: LDFPS   20(R5)          ; INITIAL FPS
                                        
021062  004714                                 JSR     PC,(R4)         ; GO PERFORM LDD/STD SEQUENCE
                                        
021064  170237  002330                         STFPS   FPS             ; STORE FPS AFTER
021070  170337  002332                         STST    FEC             ; STORE FEC/FEA AFTER
                                        
021074  023765  002330  000022                 CMP     FPS,22(R5)      ; CHECK FPS
021102  001401                                 BEQ     65$             ; FPS IS OK
021104  104077                                 ERROR   77              ; FPS BAD
021106  005765  000024                  65$:   TST     24(R5)          ; DOES FEC/FEA APPLY?
021112  100016                                 BPL     66$             ; NO - SKIP TEST
021114  010437  002344                         MOV     R4,EXPFEA       ; GET EXPECTED FEA
021120  062737  000004  002344                 ADD     #4,EXPFEA       ;   AND ADJUST
021126  123765  002332  000024                 CMPB    FEC,24(R5)      ; COMPARE FEC-S
021134  001004                                 BNE     64$             ; NOT EQUAL
021136  023737  002334  002344                 CMP     FEA,EXPFEA      ; COMPARE FEA-S
021144  001401                                 BEQ     66$             ; FEC, FEA OK
021146  104102                          64$:   ERROR   102             ; FEC OR FEA ARE BAD
021150                                  66$:
                                        
021150  023765  001170  000010                 CMP     $REG0,10(R5)    ; 1ST WORD OF RESULT CHECK?
021156  001014                                 BNE     67$             ; NO
021160  023765  001172  000012                 CMP     $REG1,12(R5)    ; 2ND WORD OF RESULT CHECK?
021166  001010                                 BNE     67$             ; NO
021170  023765  001174  000014                 CMP     $REG2,14(R5)    ; 3RD WORD OF RESULT CHECK?
021176  001004                                 BNE     67$             ; NO
021200  023765  001176  000016                 CMP     $REG3,16(R5)    ; 4TH WORD OF RESULT CHECK?
021206  001401                                 BEQ     68$             ; ALL WORDS OK
021210  104104                          67$:   ERROR   104             ; NUMBERS NOT EQUAL
021212                                  68$:
                                        
021212  000207                                 RTS     PC              ; RETURN TO TEST CALLER
                                        
                                        ; * * *  LDD/STD SEQUENCE SUBR  * * * * * * * * * * * * * * * * * *
                                        
021214  172437  002366                  LDARD0: LDD     PREVAC,AC0      ; PREV AC0 CONTENTS
021220  172415                                 LDD     (R5),AC0        ; LOAD AC0
021222  174037  001170                         STD     AC0,$REG0       ; STORE AC0
021226  000207                                 RTS     PC              ;
021230  172537  002366                  LDARD1: LDD     PREVAC,AC1      ; PREV AC1 CONTENTS
021234  172515                                 LDD     (R5),AC1        ; LOAD AC1
021236  174137  001170                         STD     AC1,$REG0       ; STORE AC1
021242  000207                                 RTS     PC              ;
021244  172637  002366                  LDARD2: LDD     PREVAC,AC2      ; PREV AC2 CONTENTS
021250  172615                                 LDD     (R5),AC2        ; LOAD AC2
```

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

```
                                        .SBTTL  SUBR TO PERFORM TEST OF LDD/LDF/STD
                                        LDAFT:
021334                                          MOV     #15,R0          ; LOAD $TMP0-14
021334  012700  000015                          MOV     R5,R1           ;   WITH TEST DATA SETS
021340  010501                          MOV     #$TMP0,R2       ;   FOR DISPLAY LATER
021342  012702  001230                  MOV     (R1)+,(R2)+     ;
021346  012122                          SOB     R0,.-2          ;
021350  077002                          MOV     #LDAFL,$LPERR   ; ERROR LOOPING ADDRESS
021352  012737  021360  001112
                                        LDAFL:  SETD                    ; D MODE FOR INITIAL LOAD
021360  170011                          JSR     PC,(R4)         ; GO PERFORM LDD/LDF/STD SEQUENCE
021362  004714                          STFPS   FPS             ; STORE FPS AFTER
021364  170237  002330                  STST    FEC             ; STORE FEC/FEA AFTER
021370  170337  002332                  CMP     FPS,26(R5)      ; CHECK FPS
021374  023765  002330  000026          BEQ     65$             ; FPS IS OK
021402  001401                          ERROR   100             ; FPS BAD
021404  104100                  65$:    TST     30(R5)          ; DOES FEC/FEA APPLY?
021406  005765  000030                  BPL     66$             ; NO - SKIP TEST
021412  100016                          MOV     R4,EXPFEA       ; GET EXPECTED FEA
021414  010437  002344                  ADD     #6,EXPFEA       ;   AND ADJUST
021420  062737  000006  002344          CMPB    FEC,30(R5)      ; COMPARE FEC-S
021426  123765  002332  000030          BNE     64$             ; NOT EQUAL
021434  001004                          CMP     FEA,EXPFEA      ; COMPARE FEA-S
021436  023737  002334  002344          BEQ     66$             ; FEC, FEA OK
021444  001401                  64$:    ERROR   103             ; FEC OR FEA ARE BAD
021446  104103                  66$:
021450
                                        CMP     $REG0,14(R5)    ; 1ST WORD OF RESULT CHECK?
021450  023765  001170  000014          BNE     67$             ; NO
021456  001014                          CMP     $REG1,16(R5)    ; 2ND WORD OF RESULT CHECK?
021460  023765  001172  000016          BNE     67$             ; NO
021466  001010                          CMP     $REG2,20(R5)    ; 3RD WORD OF RESULT CHECK?
021470  023765  001174  000020          BNE     67$             ; NO
021476  001004                          CMP     $REG3,22(R5)    ; 4TH WORD OF RESULT CHECK?
021500  023765  001176  000022          BEQ     68$             ; ALL WORDS OK
021506  001401                  67$:    ERROR   105             ; NUMBERS NOT EQUAL
021510  104105                  68$:
021512
                                        RTS     PC              ; RETURN TO TEST CALLER
021512  000207
                                        ; * * *  LDD/LDF/STD SEQUENCE SUBR  * * * * * * * * * * * * * * * * * *
                                        LDARF0: LDD     (R5),AC0        ; PREV AC0 CONTENTS
021514  172415                          LDFPS   24(R5)          ; INITIAL FPS
021516  170165  000024                  LDF     10(R5),AC0      ; LOAD NEW HALF OF AC0
021522  172465  000010                  SETD                    ; D MODE FOR STORE
021526  170011                          STD     AC0,$REG0       ; STORE ALL OF RESULT
021530  174037  001170                  RTS     PC              ;
021534  000207                  LDARF1: LDD     (R5),AC1        ; PREV AC1 CONTENTS
021536  172515                          LDFPS   24(R5)          ; INITIAL FPS
021540  170165  000024                  LDF     10(R5),AC1      ; LOAD NEW HALF OF AC1
021544  172565  000010                  SETD                    ; D MODE FOR STORE
021550  170011
```

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

```
                                        .SBTTL  SUBR TO TEST THE ABSF INSTRUCTION
                                        
021704                                  ABSFT:
021704  012700  000007                        MOV     #7,R0           ; LOAD $TMP0-6
021710  010501                                MOV     R5,R1           ;   WITH TEST DATA SETS
021712  012702  001230                      MOV     #$TMP0,R2       ;   FOR DISPLAY LATER
021716  012122                              MOV     (R1)+,(R2)+     ;
021720  077002                              SOB     R0,.-2          ;
021722  012737  021730  001112              MOV     #ABSFL,$LPERR   ; ERROR LOOPING ADDRESS
                                        
021730  170001                          ABSFL:  SETF                    ; F MODE
021732  011537  001170                      MOV     (R5),$REG0      ; INITIAL FLOAT NUMBER
021736  016537  000002  001172              MOV     2(R5),$REG1     ;   INTO OPERATING ROOM
021744  170165  000010                      LDFPS   10(R5)          ; INITIAL FPS
                                        
021750  170637  001170                  ABSFI:  ABSF    $REG0           ; ABS[($REG0)]->$REG0
                                        
021754  170237  002330                      STFPS   FPS             ; STORE FPS AFTER
021760  170337  002332                      STST    FEC             ; STORE FEC/FEA AFTER
                                        
021764  023765  002330  000012              CMP     FPS,12(R5)      ; CHECK FPS
021772  001401                              BEQ     65$             ; FPS IS OK
021774  104076                              ERROR   76              ; FPS BAD
021776  005765  000014                  65$:    TST     14(R5)          ; DOES FEC/FEA APPLY?
022002  100014                              BPL     66$             ; NO - SKIP TEST
022004  012737  021750  002344              MOV     #ABSFI,EXPFEA   ; GET EXPECTED FEA
022012  123765  002332  000014              CMPB    FEC,14(R5)      ; COMPARE FEC-S
022020  001004                              BNE     64$             ; NOT EQUAL
022022  023737  002334  002344              CMP     FEA,EXPFEA      ; COMPARE FEA-S
022030  001401                              BEQ     66$             ; FEC, FEA OK
022032  104101                          64$:    ERROR   101             ; FEC OR FEA ARE BAD
022034                                  66$:
                                        
022034  023765  001170  000004              CMP     $REG0,4(R5)     ; 1ST WORD OF RESULT CHECK?
022042  001004                              BNE     67$             ; NO
022044  023765  001172  000006              CMP     $REG1,6(R5)     ; 2ND WORD OF RESULT CHECK?
022052  001401                              BEQ     68$             ; ALL WORDS OK
022054  104106                          67$:    ERROR   106             ; NUMBERS NOT EQUAL
022056                                  68$:
                                        
022056  000207                              RTS     PC              ; RETURN TO TEST CALLER
                                        
                                        ;;*****************************************************************
                                        .SBTTL  SUBR TO TEST THE ABSD INSTRUCTION
                                        
022060                                  ABSDT:
022060  012700  000013                        MOV     #13,R0          ; LOAD $TMP0-12
022064  010501                                MOV     R5,R1           ;   WITH TEST DATA SETS
022066  012702  001230                      MOV     #$TMP0,R2       ;   FOR DISPLAY LATER
022072  012122                              MOV     (R1)+,(R2)+     ;
022074  077002                              SOB     R0,.-2          ;
022076  012737  022104  001112              MOV     #ABSDL,$LPERR   ; ERROR LOOPING ADDRESS
                                        
022104  170011                          ABSDL:  SETD                    ; D MODE
022106  011537  001170                      MOV     (R5),$REG0      ; INITIAL FLOAT NUMBER
022112  016537  000002  001172              MOV     2(R5),$REG1     ;   INTO OPERATING ROOM
```

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

```
                                        .SBTTL  SUBR TO TEST THE NEGF INSTRUCTION
                                
022270                          NEGFT:
022270  012700  000007                  MOV     #7,R0           ; LOAD $TMP0-6
022274  010501                          MOV     R5,R1           ;   WITH TEST DATA SETS
022276  012702  001230                  MOV     #$TMP0,R2       ;   FOR DISPLAY LATER
022302  012122                          MOV     (R1)+,(R2)+     ;
022304  077002                          SOB     R0,.-2          ;
022306  012737  022314  001112          MOV     #NEGFL,$LPERR   ; ERROR LOOPING ADDRESS
                                
022314  170001                  NEGFL:  SETF                    ; F MODE
022316  011537  001170                  MOV     (R5),$REG0      ; INITIAL FLOAT NUMBER
022322  016537  000002  001172          MOV     2(R5),$REG1     ;   INTO OPERATING ROOM
022330  170165  000010                  LDFPS   10(R5)          ; INITIAL FPS
                                
022334  170737  001170          NEGFI:  NEGF    $REG0           ; -($REG0)->$REG0
                                
022340  170237  002330                  STFPS   FPS             ; STORE FPS AFTER
022344  170337  002332                  STST    FEC             ; STORE FEC/FEA AFTER
                                
022350  023765  002330  000012          CMP     FPS,12(R5)      ; CHECK FPS
022356  001401                          BEQ     65$             ; FPS IS OK
022360  104076                          ERROR   76              ; FPS BAD
022362  005765  000014          65$:    TST     14(R5)          ; DOES FEC/FEA APPLY?
022366  100014                          BPL     66$             ; NO - SKIP TEST
022370  012737  022334  002344          MOV     #NEGFI,EXPFEA   ; GET EXPECTED FEA
022376  123765  002332  000014          CMPB    FEC,14(R5)      ; COMPARE FEC-S
022404  001004                          BNE     64$             ; NOT EQUAL
022406  023737  002334  002344          CMP     FEA,EXPFEA      ; COMPARE FEA-S
022414  001401                          BEQ     66$             ; FEC, FEA OK
022416  104101                  64$:    ERROR   101             ; FEC OR FEA ARE BAD
022420                          66$:
                                
022420  023765  001170  000004          CMP     $REG0,4(R5)     ; 1ST WORD OF RESULT CHECK?
022426  001004                          BNE     67$             ; NO
022430  023765  001172  000006          CMP     $REG1,6(R5)     ; 2ND WORD OF RESULT CHECK?
022436  001401                          BEQ     68$             ; ALL WORDS OK
022440  104110                  67$:    ERROR   110             ; NUMBERS NOT EQUAL
022442                          68$:
                                
022442  000207                          RTS     PC              ; RETURN TO TEST CALLER
                                
                                ;;*****************************************************************
                                .SBTTL  SUBR TO TEST THE NEGD INSTRUCTION
                                
022444                          NEGDT:
022444  012700  000013                  MOV     #13,R0          ; LOAD $TMP0-12
022450  010501                          MOV     R5,R1           ;   WITH TEST DATA SETS
022452  012702  001230  ;               MOV     #$TMP0,R2       ;   FOR DISPLAY LATER
022456  012122                          MOV     (R1)+,(R2)+     ;
022460  077002                          SOB     R0,.-2          ;
022462  012737  022470  001112          MOV     #NEGDL,$LPERR   ; ERROR LOOPING ADDRESS
                                
022470  170011                  NEGDL:  SETD                    ; D MODE
022472  011537  001170                  MOV     (R5),$REG0      ; INITIAL FLOAT NUMBER
022476  016537  000002  001172          MOV     2(R5),$REG1     ;   INTO OPERATING ROOM
```

```
4681  022504  016537  000004  001174          MOV     4(R5),$REG2     ;
4682  022512  016537  000006  001176          MOV     6(R5),$REG3     ;
4683  022520  170165  000020                  LDFPS   20(R5)          ; INITIAL FPS
4684
4685  022524  170737  001170          NEGDI:  NEGD    $REG0           ; -($REG0)->$REG0
4686
4687  022530  170237  002330                  STFPS   FPS             ; STORE FPS AFTER
4688  022534  170337  002332                  STST    FEC             ; STORE FEC/FEA AFTER
4689
4690  022540  023765  002330  000022          CMP     FPS,22(R5)      ; CHECK FPS
4691  022546  001401                          BEQ     65$             ; FPS IS OK
4692  022550  104077                          ERROR   77              ; FPS BAD
4693  022552  005765  000024          65$:    TST     24(R5)          ; DOES FEC/FEA APPLY?
4694  022556  100014                          BPL     66$             ; NO - SKIP TEST
4695  022560  012737  022524  002344          MOV     #NEGDI,EXPFEA   ; GET EXPECTED FEA
4696  022566  123765  002332  000024          CMPB    FEC,24(R5)      ; COMPARE FEC-S
4697  022574  001004                          BNE     64$             ; NOT EQUAL
4698  022576  023737  002334  002344          CMP     FEA,EXPFEA      ; COMPARE FEA-S
4699  022604  001401                          BEQ     66$             ; FEC, FEA OK
4700  022606  104102                  64$:    ERROR   102             ; FEC OR FEA ARE BAD
4701  022610                          66$:
4702
4703  022610  023765  001170  000010          CMP     $REG0,10(R5)    ; 1ST WORD OF RESULT CHECK?
4704  022616  001014                          BNE     67$             ; NO
4705  022620  023765  001172  000012          CMP     $REG1,12(R5)    ; 2ND WORD OF RESULT CHECK?
4706  022626  001010                          BNE     67$             ; NO
4707  022630  023765  001174  000014          CMP     $REG2,14(R5)    ; 3RD WORD OF RESULT CHECK?
4708  022636  001004                          BNE     67$             ; NO
4709  022640  023765  001176  000016          CMP     $REG3,16(R5)    ; 4TH WORD OF RESULT CHECK?
4710  022646  001401                          BEQ     68$             ; ALL WORDS OK
4711  022650  104111                  67$:    ERROR   111             ; NUMBERS NOT EQUAL
4712  022652                          68$:
4713
4714  022652  000207                          RTS     PC              ; RETURN TO TEST CALLER
4715
```

```
                                        .SBTTL  SUBR TO TEST THE CLRF INSTRUCTION

                                CLRFT:
                                        MOV     #7,R0           ; LOAD $TMP0-6
                                        MOV     R5,R1           ;   WITH TEST DATA SETS
                                        MOV     #$TMP0,R2       ;   FOR DISPLAY LATER
                                        MOV     (R1)+,(R2)+     ;
                                        SOB     R0,.-2          ;
                                        MOV     #CLRFL,$LPERR   ; ERROR LOOPING ADDRESS

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

                                CLRFI:  CLRF    $REG0           ; 0->$REG0

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

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

                                        CMP     $REG0,4(R5)     ; 1ST WORD OF RESULT CHECK?
                                        BNE     67$             ; NO
                                        CMP     $REG1,6(R5)     ; 2ND WORD OF RESULT CHECK?
                                        BEQ     68$             ; ALL WORDS OK
                                67$:    ERROR   112             ; NUMBERS NOT EQUAL
                                68$:

                                        RTS     PC              ; RETURN TO TEST CALLER

                                ;;*****************************************************************
                                .SBTTL  SUBR TO TEST THE CLRD INSTRUCTION

                                CLRDT:
                                        MOV     #13,R0          ; LOAD $TMP0-12
                                        MOV     R5,R1           ;   WITH TEST DATA SETS
                                        MOV     #$TMP0,R2       ;   FOR DISPLAY LATER
                                        MOV     (R1)+,(R2)+     ;
                                        SOB     R0,.-2          ;
                                        MOV     #CLRDL,$LPERR   ; ERROR LOOPING ADDRESS

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

```
4772  023070  016537  000004  001174          MOV     4(R5),$REG2     ;
4773  023076  016537  000006  001176          MOV     6(R5),$REG3     ;
4774  023104  170165  000020                  LDFPS   20(R5)          ; INITIAL FPS
4775
4776  023110  170437  001170          CLRDI:  CLRD    $REG0           ; 0->$REG0
4777
4778  023114  170237  002330                  STFPS   FPS             ; STORE FPS AFTER
4779  023120  170337  002332                  STST    FEC             ; STORE FEC/FEA AFTER
4780
4781  023124  023765  002330  000022          CMP     FPS,22(R5)      ; CHECK FPS
4782  023132  001401                          BEQ     65$             ; FPS IS OK
4783  023134  104077                          ERROR   77              ; FPS BAD
4784  023136  005765  000024          65$:    TST     24(R5)          ; DOES FEC/FEA APPLY?
4785  023142  100014                          BPL     66$             ; NO - SKIP TEST
4786  023144  012737  023110  002344          MOV     #CLRDI,EXPFEA   ; GET EXPECTED FEA
4787  023152  123765  002332  000024          CMPB    FEC,24(R5)      ; COMPARE FEC-S
4788  023160  001004                          BNE     64$             ; NOT EQUAL
4789  023162  023737  002334  002344          CMP     FEA,EXPFEA      ; COMPARE FEA-S
4790  023170  001401                          BEQ     66$             ; FEC, FEA OK
4791  023172  104102                  64$:    ERROR   102             ; FEC OR FEA ARE BAD
4792  023174                          66$:
4793
4794  023174  023765  001170  000010          CMP     $REG0,10(R5)    ; 1ST WORD OF RESULT CHECK?
4795  023202  001014                          BNE     67$             ; NO
4796  023204  023765  001172  000012          CMP     $REG1,12(R5)    ; 2ND WORD OF RESULT CHECK?
4797  023212  001010                          BNE     67$             ; NO
4798  023214  023765  001174  000014          CMP     $REG2,14(R5)    ; 3RD WORD OF RESULT CHECK?
4799  023222  001004                          BNE     67$             ; NO
4800  023224  023765  001176  000016          CMP     $REG3,16(R5)    ; 4TH WORD OF RESULT CHECK?
4801  023232  001401                          BEQ     68$             ; ALL WORDS OK
4802  023234  104113                  67$:    ERROR   113             ; NUMBERS NOT EQUAL
4803  023236                          68$:
4804
4805  023236  000207                          RTS     PC              ; RETURN TO TEST CALLER
4806
4807
```

```
                                        .SBTTL  SUBR TO TEST THE TSTF INSTRUCTION
                                        TSTFT:
023240                                          MOV     #7,R0           ; LOAD $TMP0-6
023240  012700  000007                          MOV     R5,R1           ;   WITH TEST DATA SETS
023244  010501                          MOV     #$TMP0,R2       ;   FOR DISPLAY LATER
023246  012702  001230                  MOV     (R1)+,(R2)+     ;
023252  012122                          SOB     R0,.-2          ;
023254  077002                          MOV     #TSTFL,$LPERR   ; ERROR LOOPING ADDRESS
023256  012737  023264  001112
                                        TSTFL:  SETF                    ; F MODE
023264  170001                          MOV     (R5),$REG0      ; INITIAL FLOAT NUMBER
023266  011537  001170                  MOV     2(R5),$REG1     ;   INTO OPERATING ROOM
023272  016537  000002  001172          LDFPS   10(R5)          ; INITIAL FPS
023300  170165  000010
                                        TSTFI:  TSTF    $REG0           ; ($REG0)->$REG0
023304  170537  001170
                                        STFPS   FPS             ; STORE FPS AFTER
023310  170237  002330                  STST    FEC             ; STORE FEC/FEA AFTER
023314  170337  002332
                                        CMP     FPS,12(R5)      ; CHECK FPS
023320  023765  002330  000012          BEQ     65$             ; FPS IS OK
023326  001401                          ERROR   76              ; FPS BAD
023330  104076                  65$:    TST     14(R5)          ; DOES FEC/FEA APPLY?
023332  005765  000014                  BPL     66$             ; NO - SKIP TEST
023336  100014                          MOV     #TSTFI,EXPFEA   ; GET EXPECTED FEA
023340  012737  023304  002344          CMPB    FEC,14(R5)      ; COMPARE FEC-S
023346  123765  002332  000014          BNE     64$             ; NOT EQUAL
023354  001004                          CMP     FEA,EXPFEA      ; COMPARE FEA-S
023356  023737  002334  002344          BEQ     66$             ; FEC, FEA OK
023364  001401                  64$:    ERROR   101             ; FEC OR FEA ARE BAD
023366  104101                  66$:
023370
                                        CMP     $REG0,4(R5)     ; 1ST WORD OF RESULT CHECK?
023370  023765  001170  000004          BNE     67$             ; NO
023376  001004                          CMP     $REG1,6(R5)     ; 2ND WORD OF RESULT CHECK?
023400  023765  001172  000006          BEQ     68$             ; ALL WORDS OK
023406  001401                  67$:    ERROR   114             ; NUMBERS NOT EQUAL
023410  104114                  68$:
023412
                                        RTS     PC              ; RETURN TO TEST CALLER
023412  000207
                                        ;;*****************************************************************
                                        .SBTTL  SUBR TO TEST THE TSTD INSTRUCTION

023414                                  TSTDT:
023414  012700  000013                  MOV     #13,R0          ; LOAD $TMP0-12
023420  010501                          MOV     R5,R1           ;   WITH TEST DATA SETS
023422  012702  001230                  MOV     #$TMP0,R2       ;   FOR DISPLAY LATER
023426  012122                          MOV     (R1)+,(R2)+     ;
023430  077002                          SOB     R0,.-2          ;
023432  012737  023440  001112          MOV     #TSTDL,$LPERR   ; ERROR LOOPING ADDRESS

023440  170011                  TSTDL:  SETD                    ; D MODE
023442  011537  001170                  MOV     (R5),$REG0      ; INITIAL FLOAT NUMBER
023446  016537  000002  001172          MOV     2(R5),$REG1     ;   INTO OPERATING ROOM
```

```
4864  023454  016537  000004  001174          MOV     4(R5),$REG2     ;
4865  023462  016537  000006  001176          MOV     6(R5),$REG3     ;
4866  023470  170165  000020                  LDFPS   20(R5)          ; INITIAL FPS
4867
4868  023474  170537  001170          TSTDI:  TSTD    $REG0           ; ($REG0)->$REG0
4869
4870  023500  170237  002330                  STFPS   FPS             ; STORE FPS AFTER
4871  023504  170337  002332                  STST    FEC             ; STORE FEC/FEA AFTER
4872
4873  023510  023765  002330  000022          CMP     FPS,22(R5)      ; CHECK FPS
4874  023516  001401                          BEQ     65$             ; FPS IS OK
4875  023520  104077                          ERROR   77              ; FPS BAD
4876  023522  005765  000024          65$:    TST     24(R5)          ; DOES FEC/FEA APPLY?
4877  023526  100014                          BPL     66$             ; NO - SKIP TEST
4878  023530  012737  023474  002344          MOV     #TSTDI,EXPFEA   ; GET EXPECTED FEA
4879  023536  123765  002332  000024          CMPB    FEC,24(R5)      ; COMPARE FEC-S
4880  023544  001004                          BNE     64$             ; NOT EQUAL
4881  023546  023737  002334  002344          CMP     FEA,EXPFEA      ; COMPARE FEA-S
4882  023554  001401                          BEQ     66$             ; FEC, FEA OK
4883  023556  104102                  64$:    ERROR   102             ; FEC OR FEA ARE BAD
4884  023560                          66$:
4885
4886  023560  023765  001170  000010          CMP     $REG0,10(R5)    ; 1ST WORD OF RESULT CHECK?
4887  023566  001014                          BNE     67$             ; NO
4888  023570  023765  001172  000012          CMP     $REG1,12(R5)    ; 2ND WORD OF RESULT CHECK?
4889  023576  001010                          BNE     67$             ; NO
4890  023600  023765  001174  000014          CMP     $REG2,14(R5)    ; 3RD WORD OF RESULT CHECK?
4891  023606  001004                          BNE     67$             ; NO
4892  023610  023765  001176  000016          CMP     $REG3,16(R5)    ; 4TH WORD OF RESULT CHECK?
4893  023616  001401                          BEQ     68$             ; ALL WORDS OK
4894  023620  104115                  67$:    ERROR   115             ; NUMBERS NOT EQUAL
4895  023622                          68$:
4896
4897  023622  000207                          RTS     PC              ; RETURN TO TEST CALLER
4898
```

```
                                        .SBTTL  FPP TRAP CATCHER
4899
4900
4901  023624  010637  002342  FPPILT: MOV     SP,FPPOSP       ; GET NEW SP
4902  023630  012637  002336          MOV     (SP)+,FPPOPC    ; POP OLD PC FOR DISPLAY
4903  023634  012637  002340          MOV     (SP)+,FPPOPS    ; POP OLD PS FOR DISPLAY
4904  023640  170237  002330          STFPS   FPS             ; GET FPS
4905  023644  170337  002332          STST    FEC             ; GET FEC/FEA
4906  023650  104116                  ERROR   116             ; SIGNAL UNEXPECTED FPP TRAP
4907  023652  013746  002340          MOV     FPPOPS,-(SP)    ; PUSH PSW
4908  023656  013746  002336          MOV     FPPOPC,-(SP)    ; PUSH PC
4909  023662  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

023664                          $SCOPE:
023664                          64$:
023664  032777  040000  155252  1$:     BIT     #BIT14,@SWR     ;;LOOP ON PRESENT TEST?
023672  001114                          BNE     $OVER           ;;YES IF SW14=1
                                ;*****START OF CODE FOR THE XOR TESTER*****
023674  000416                  $XTSTR: BR      6$              ;;IF RUNNING ON THE "XOR" TESTER CHANGE
                                                                ;;THIS INSTRUCTION TO A "NOP" (NOP=240)
023676  013746  000004                  MOV     @#ERRVEC,-(SP)  ;;SAVE THE CONTENTS OF THE ERROR VECTOR
023702  012737  023722  000004          MOV     #5$,@#ERRVEC    ;;SET FOR TIMEOUT
023710  005737  177060                  TST     @#177060        ;;TIME OUT ON XOR?
023714  012637  000004                  MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
023720  000463                          BR      $SVLAD          ;;GO TO THE NEXT TEST
023722  022626                  5$:     CMP     (SP)+,(SP)+     ;;CLEAR THE STACK AFTER A TIME OUT
023724  012637  000004                  MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
023730  000423                          BR      7$              ;;LOOP ON THE PRESENT TEST
023732                          6$:;*****END OF CODE FOR THE XOR TESTER*****
023732  032777  000400  155204          BIT     #BIT08,@SWR     ;;LOOP ON SPEC. TEST?
023740  001404                          BEQ     2$              ;;BR IF NO
023742  023737  001150  001102          CMP     $LPTST,$TSTNM   ;;ON THE RIGHT TEST?
023750  001465                          BEQ     $OVER           ;;BR IF YES
023752  005737  001104          2$:     TST     $ERFLG          ;;HAS AN ERROR OCCURRED?
023756  001421                          BEQ     3$              ;;BR IF NO
023760  023737  001120  001104          CMP     $ERMAX,$ERFLG   ;;MAX. ERRORS FOR THIS TEST OCCURRED?
023766  101015                          BHI     3$              ;;BR IF NO
023770  032777  001000  155146          BIT     #BIT09,@SWR     ;;LOOP ON ERROR?
023776  001404                          BEQ     4$              ;;BR IF NO
024000  013737  001112  001110  7$:     MOV     $LPERR,$LPADR   ;;SET LOOP ADDRESS TO LAST SCOPE
024006  000446                          BR      $OVER
024010  005037  001104          4$:     CLR     $ERFLG          ;;ZERO THE ERROR FLAG
024014  005037  001310                  CLR     $TIMES          ;;CLEAR THE NUMBER OF ITERATIONS TO MAKE
024020  000415                          BR      1$              ;;ESCAPE TO THE NEXT TEST
024022  032777  004000  155114  3$:     BIT     #BIT11,@SWR     ;;INHIBIT ITERATIONS?
024030  001011                          BNE     1$              ;;BR IF YES
024032  005737  001332                  TST     $PASS           ;;IF FIRST PASS OF PROGRAM
024036  001406                          BEQ     1$              ;;      INHIBIT ITERATIONS
024040  005237  001106                  INC     $ICNT           ;;INCREMENT ITERATION COUNT
024044  023737  001310  001106          CMP     $TIMES,$ICNT    ;;CHECK THE NUMBER OF ITERATIONS MADE
024052  002024                          BGE     $OVER           ;;BR IF MORE ITERATION REQUIRED
024054  012737  000001  001106  1$:     MOV     #1,$ICNT        ;;REINITIALIZE THE ITERATION COUNTER
024062  013737  024140  001310          MOV     $MXCNT,$TIMES   ;;SET NUMBER OF ITERATIONS TO DO
024070  005237  001102          $SVLAD: INC     $TSTNM          ;;COUNT TEST NUMBERS
024074  013737  001102  001330          MOV     $TSTNM,$TESTN   ;;SET TEST NUMBER IN APT MAILBOX
024102  011637  001110                  MOV     (SP),$LPADR     ;;SAVE SCOPE LOOP ADDRESS
```

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

```
                                .SBTTL  ERROR HANDLER ROUTINE

                                ;;*****************************************************************
                                ;*THIS ROUTINE WILL INCREMENT THE ERROR FLAG AND THE ERROR COUNT,
                                ;*SAVE THE ERROR ITEM NUMBER AND THE ADDRESS OF THE ERROR CALL
                                ;*AND GO TO $TYPERR ON ERROR
                                ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
                                ;*SW15=1         HALT ON ERROR
                                ;*SW'3=1         INHIBIT ERROR TYPEOUTS
                                ;*SW10=1         BELL ON ERROR
                                ;*SW09=1         LOOP ON ERROR
                                ;*CALL
                                ;*      ERROR   N       ;;ERROR=EMT AND N=ERROR ITEM NUMBER

024142                          $ERROR:
024142  010037  002346                  MOV     R0,EREG0        ; DISPLAY R0
024146  010137  002350                  MOV     R1,EREG1        ;         R1
024152  010237  002352                  MOV     R2,EREG2        ;         R2
024156  010337  002354                  MOV     R3,EREG3        ;         R3
024162  010437  002356                  MOV     R4,EREG4        ;         R4
024166  010537  002360                  MOV     R5,EREG5        ;         R5
024172  010637  002362                  MOV     R6,EREG6        ; GET R6(SP) BEFORE TRAP
024176  062737  000004  002362          ADD     #4,EREG6        ;
024204  011637  002364                  MOV     (SP),EREG7      ; PC -> ERROR CALL INSTR
024210  005237  001104          7$:     INC     $ERFLG          ;;SET THE ERROR FLAG
024214  001775                  BEQ     7$              ;;DON'T LET THE FLAG GO TO ZERO
024216  013777  001102  154722  MOV     $TSTNM,@DISPLAY ;;DISPLAY TEST NUMBER
024224  032777  002000  154712  BIT     #BIT10,@SWR     ;;BELL ON ERROR?
024232  001402                  BEQ     1$              ;;NO - SKIP
024234  104401  001314          TYPE    ,$BELL          ;;RING BELL
024240  005237  001114          1$:     INC     $ERTTL          ;;COUNT THE NUMBER OF ERRORS
024244  011637  001122          MOV     (SP),$ERRPC     ;;GET ADDRESS OF ERROR INSTRUCTION
024250  162737  000002  001122  SUB     #2,$ERRPC
024256  117737  154640  001116  MOVB    @$ERRPC,$ITEMB  ;;STRIP AND SAVE THE ERROR ITEM CODE
024264  032777  020000  154652  BIT     #BIT13,@SWR     ;;SKIP TYPEOUT IF SET
024272  001004                  BNE     20$             ;;SKIP TYPEOUTS
024274  004737  024404          JSR     PC,$TYPERR      ;;GO TO USER ERROR ROUTINE
024300  104401  001321          TYPE    ,$CRLF
024304                          20$:
024304  122737  000001  001344  CMPB    #APTENV,$ENV    ;;RUNNING IN APT MODE
024312  001007                  BNE     2$              ;;NO,SKIP APT ERROR REPORT
024314  113737  001116  024326  MOVB    $ITEMB,21$      ;;SET ITEM NUMBER AS ERROR NUMBER
024322  004737  025120          JSR     PC,$ATY4        ;;REPORT FATAL ERROR TO APT
024326     000                  21$:    .BYTE   0
024327     000                  .BYTE   0
024330  000777                  22$:    BR      22$             ;;APT ERROR LOOP
024332  005777  154606          2$:     TST     @SWR            ;;HALT ON ERROR
024336  100001                  BPL     3$              ;;SKIP IF CONTINUE
024340  000000                  HALT                    ;;HALT ON ERROR!
024342  032777  001000  154574  3$:     BIT     #BIT09,@SWR     ;;LOOP ON ERROR SWITCH SET?
024350  001402                  BEQ     4$              ;;BR IF NO
024352  013716  001112          MOV     $LPERR,(SP)     ;;FUDGE RETURN FOR LOOPING
024356  005737  001312          4$:     TST     $ESCAPE         ;;CHECK FOR AN ESCAPE ADDRESS
024362  001402                  BEQ     5$              ;;BR IF NONE
024364  013716  001312          MOV     $ESCAPE,(SP)    ;;FUDGE RETURN ADDRESS FOR ESCAPE
024370                          5$:
```

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

```
                                ;;*****************************************************************

                                .SBTTL  ERROR MESSAGE TYPEOUT ROUTINE (MODIFIED SYSMAC)

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

024404                          $TYPERR:
024404  104401                          TYPE                    ; TYPE "HOT" OR "WARM"
024406  001321                  HOTWRM: .WORD   $CRLF            ; PTR TO MESSAGE
024410  010046                          MOV     R0,-(SP)        ; SAVE R0
024412  010146                          MOV     R1,-(SP)        ; SAVE R1
024414  005000                          CLR     R0              ; PICKUP ITEM INDEX
024416  153700  001116                  BISB    @#$ITEMB,R0     ;
024422  001004                          BNE     1$              ; IF ITEM NUMBER FROM ERROR 0,
                                                                ;   JUST TYPE PC OF ERROR
024424  013746  001122                  MOV     $ERRPC,-(SP)    ; GET ERROR PC FOR TYPEOUT
024430  104402                          TYPOC                   ; TYPE OCTAL, ALL DIGITS
024432  000452                          BR      7$              ; EXIT
024434  005300                  1$:     DEC     R0              ; ADJUST ERROR # FOR TABLE INDEX
024436  006300                          ASL     R0              ;   OF 6 BYTES/ENTRY
024440  010001                          MOV     R0,R1           ;
024442  006300                          ASL     R0              ;
024444  060100                          ADD     R1,R0           ;
024446  062700  001354                  ADD     #$ERRTB,R0      ; FORM TABLE PTR
024452  012037  024462                  MOV     (R0)+,2$        ; PICKUP "ERROR MESSAGE" PTR
024456  001404                          BEQ     3$              ; SKIP TYPEOUT IF NULL
024460  104401                          TYPE                    ; TYPE "ERROR MESSAGE"
024462  000000                  2$:     .WORD   0               ; "ERROR MESSAGE" PTR HERE
024464  104401  001321                  TYPE    ,$CRLF          ; CR & LF
024470  104401  024600          3$:     TYPE    ,11$            ; "TEST #    ERR PC" HEADER
024474  012037  024504                  MOV     (R0)+,4$        ; PICKUP "DATA HEADER" PTR
024500  001402                          BEQ     5$              ; SKIP TYPEOUT IF NULL
024502  104401                          TYPE                    ; TYPE "DATA HEADER"
024504  000000                  4$:     .WORD   0               ; "DATA HEADER" PTR HERE
024506  104401  001321          5$:     TYPE    ,$CRLF          ; CR & LF
024512  017746  000054                  MOV     @8$,-(SP)       ; ($TESTN)
024516  104402                          TYPOC                   ; OCTAL W/ LEADING ZEROS
024520  104401  024576                  TYPE    ,10$            ; <HT>
024524  017746  000044                  MOV     @9$,-(SP)       ; ($ERRPC)
024530  104402                          TYPOC                   ; OCTAL W/ LEADING ZEROS
024532  104401  024576                  TYPE    ,10$            ; <HT>
024536  011000                          MOV     (R0),R0         ; PICKUP "DATA TABLE" PTR
024540  001407                          BEQ     7$              ; EXIT IF NULL
024542  013046                  6$:     MOV     @(R0)+,-(SP)    ; SAVE ... FOR TYPEOUT
024544  104402                          TYPOC                   ; TYPE OCTAL, ALL DIGITS
024546  005710                          TST     (R0)            ; ANOTHER NUMBER ?
024550  001403                          BEQ     7$              ; NO - EXIT
024552  104401  024576                  TYPE    ,10$            ; TAB BETWEEN ELEMENTS
```

```
5090  024556  000771                        BR      6$              ; LOOP ON DATA TABLE VECTOR
5091  024560  012601                7$:     MOV     (SP)+,R1        ; RESTORE R1
5092  024562  012600                        MOV     (SP)+,R0        ; RESTORE R0
5093  024564  104401  001321                TYPE    ,$CRLF          ; CR & LF
5094  024570  000207                        RTS     PC              ; RETURN
5095  024572  001330                8$:     .WORD   $TESTN          ;
5096  024574  001122                9$:     .WORD   $ERRPC          ;
5097  024576  000011                10$:    .ASCIZ  <11>            ; <HT>
5098  024600  042524  052123  021440 11$:    .ASCIZ  "TEST # ERR PC  "
5099  024606  042411  051122  050040
5100  024614  004503     000
5101          024620                        .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
                                ;*

024620  105737  001165          $TYPE:  TSTB    $TPFLG          ;;IS THERE A TERMINAL?
024624  100002                          BPL     1$              ;;BR IF YES
024626  000000                          HALT                    ;;HALT HERE IF NO TERMINAL
024630  000430                          BR      3$              ;;LEAVE
024632  010046                  1$:     MOV     R0,-(SP)        ;;SAVE R0
024634  017600  000002                  MOV     2(SP),R0        ;;GET ADDRESS OF ASCIZ STRING
024640  122737  000001  001344          CMPB    #APTENV,$ENV    ;;RUNNING IN APT MODE
024646  001011                          BNE     62$             ;;NO,GO CHECK FOR APT CONSOLE
024650  132737  000100  001345          BITB    #APTSPOOL,$ENVM ;;SPOOL MESSAGE TO APT
024656  001405                          BEQ     62$             ;;NO,GO CHECK FOR CONSOLE
024660  010037  024670                  MOV     R0,61$          ;;SETUP MESSAGE ADDRESS FOR APT
024664  004737  025110                  JSR     PC,$ATY3        ;;SPOOL MESSAGE TO APT
024670  000000                  61$:    .WORD   0               ;;MESSAGE ADDRESS
024672  132737  000040  001345  62$:    BITB    #APTCSUP,$ENVM  ;;APT CONSOLE SUPPRESSED
024700  001003                          BNE     60$             ;;YES,SKIP TYPE OUT
024702  112046                  2$:     MOVB    (R0)+,-(SP)     ;;PUSH CHARACTER TO BE TYPED ONTO STACK
024704  001005                          BNE     4$              ;;BR IF IT ISN'T THE TERMINATOR
024706  005726                          TST     (SP)+           ;;IF TERMINATOR POP IT OFF THE STACK
024710  012600                  60$:    MOV     (SP)+,R0        ;;RESTORE R0
024712  062716  000002          3$:     ADD     #2,(SP)         ;;ADJUST RETURN PC
024716  000002                          RTI                     ;;RETURN
024720  122716  000011          4$:     CMPB    #HT,(SP)        ;;BRANCH IF <HT>
024724  001430                          BEQ     8$
024726  122716  000200                  CMPB    #CRLF,(SP)      ;;BRANCH IF NOT <CRLF>
024732  001006                          BNE     5$
024734  005726                          TST     (SP)+           ;;POP  <CR><LF> EQUIV
024736  104401                          TYPE                    ;;TYPE A CR AND LF
024740  001321                          $CRLF
024742  105037  025076                  CLRB    $CHARCNT        ;;CLEAR CHARACTER COUNT
024746  000755                          BR      2$              ;;GET NEXT CHARACTER
024750  004737  025032          5$:     JSR     PC,$TYPEC       ;;GO TYPE THIS CHARACTER
024754  123726  001164          6$:     CMPB    $FILLC,(SP)+    ;;IS IT TIME FOR FILLER CHARS.?
024760  001350                          BNE     2$              ;;IF NO GO GET NEXT CHAR.
024762  013746  001162                  MOV     $NULL,-(SP)     ;;GET # OF FILLER CHARS. NEEDED
                                                                ;;AND THE NULL CHAR.
024766  105366  000001          7$:     DECB    1(SP)           ;;DOES A NULL NEED TO BE TYPED?
024772  002770                          BLT     6$              ;;BR IF NO--GO POP THE NULL OFF OF STACK
024774  004737  025032                  JSR     PC,$TYPEC       ;;GO TYPE A NULL
025000  105337  025076                  DECB    $CHARCNT        ;;DO NOT COUNT AS A COUNT
```

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

```
                                .SBTTL  APT COMMUNICATIONS ROUTINE
                                ;;*****************************************************************
025102  112737  000001  025346  $ATY1:  MOVB    #1,$FFLG        ;;TO REPORT FATAL ERROR
025110  112737  000001  025344  $ATY3:  MOVB    #1,$MFLG        ;;TO TYPE A MESSAGE
025116  000403                        BR      $ATYC
025120  112737  000001  025346  $ATY4:  MOVB    #1,$FFLG        ;;TO ONLY REPORT FATAL ERROR
025126                          $ATYC:
025126  010046                        MOV     R0,-(SP)        ;;PUSH R0 ON STACK
025130  010146                        MOV     R1,-(SP)        ;;PUSH R1 ON STACK
025132  105737  025344                TSTB    $MFLG           ;;SHOULD TYPE A MESSAGE?
025136  001450                        BEQ     5$              ;;IF NOT:  BR
025140  122737  000001  001344        CMPB    #APTENV,$ENV    ;;OPERATING UNDER APT?
025146  001031                        BNE     3$              ;;IF NOT:  BR
025150  132737  000100  001345        BITB    #APTSPOOL,$ENVM ;;SHOULD SPOOL MESSAGES?
025156  001425                        BEQ     3$              ;;IF NOT:  BR
025160  017600  000004                MOV     @4(SP),R0       ;;GET MESSAGE ADDR.
025164  062766  000002  000004        ADD     #2,4(SP)                ;;BUMP RETURN ADDR.
025172  005737  001324          1$:   TST     $MSGTYPE        ;;SEE IF DONE W/ LAST XMISSION?
025176  001375                        BNE     1$              ;;IF NOT:  WAIT
025200  010037  001340                MOV     R0,$MSGAD       ;;PUT ADDR IN MAILBOX
025204  105720                  2$:   TSTB    (R0)+           ;;FIND END OF MESSAGE
025206  001376                        BNE     2$
025210  163700  001340                SUB     $MSGAD,R0       ;;SUB START OF MESSAGE
025214  006200                        ASR     R0              ;;GET MESSAGE LNGTH IN WORDS
025216  010037  001342                MOV     R0,$MSGLGT      ;;PUT LENGTH IN MAILBOX
025222  012737  000004  001324        MOV     #4,$MSGTYPE     ;;TELL APT TO TAKE MSG.
025230  000413                        BR      5$
025232  017637  000004  025256  3$:   MOV     @4(SP),4$       ;;PUT MSG ADDR IN JSR LINKAGE
025240  062766  000002  000004        ADD     #2,4(SP)                ;;BUMP RETURN ADDRESS
025246  013746  177776                MOV     177776,-(SP)    ;;PUSH 177776 ON STACK
025252  004737  024620                JSR     PC,$TYPE        ;;CALL TYPE MACRO
025256  000000                  4$:   .WORD   0
025260                          5$:
025260  105737  025346          10$:  TSTB    $FFLG           ;;SHOULD REPORT FATAL ERROR?
025264  001416                        BEQ     12$             ;;IF NOT:  BR
025266  005737  001344                TST     $ENV            ;;RUNNING UNDER APT?
025272  001413                        BEQ     12$             ;;IF NOT:  BR
025274  005737  001324          11$:  TST     $MSGTYPE        ;;FINISHED LAST MESSAGE?
025300  001375                        BNE     11$             ;;IF NOT:  WAIT
025302  017637  000004  001326        MOV     @4(SP),$FATAL   ;;GET ERROR #
025310  062766  000002  000004        ADD     #2,4(SP)                ;;BUMP RETURN ADDR.
025316  005237  001324                INC     $MSGTYPE        ;;TELL APT TO TAKE ERROR
025322  105037  025346          12$:  CLRB    $FFLG           ;;CLEAR FATAL FLAG
025326  105037  025345                CLRB    $LFLG           ;;CLEAR LOG FLAG
025332  105037  025344                CLRB    $MFLG           ;;CLEAR MESSAGE FLAG
025336  012601                        MOV     (SP)+,R1        ;;POP STACK INTO R1
025340  012600                        MOV     (SP)+,R0        ;;POP STACK INTO R0
025342  000207                        RTS     PC              ;;RETURN
025344     000          $MFLG:  .BYTE   0               ;;MESSG. FLAG
025345     000          $LFLG:  .BYTE   0               ;;LOG FLAG
025346     000          $FFLG:  .BYTE   0               ;;FATAL FLAG
025350                          .EVEN
000200                  APTSIZE=200
000001                  APTENV=001
000100                  APTSPOOL=100
```

```
  5237          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

025350  017646  000000          $TYPOS: MOV     @(SP),-(SP)     ;;PICKUP THE MODE
025354  116637  000001  025573          MOVB    1(SP),$OFILL    ;;LOAD ZERO FILL SWITCH
025362  112637  025575          MOVB    (SP)+,$OMODE+1  ;;NUMBER OF DIGITS TO TYPE
025366  062716  000002          ADD     #2,(SP)         ;;ADJUST RETURN ADDRESS
025372  000406                  BR      $TYPON
025374  112737  000001  025573  $TYPOC: MOVB    #1,$OFILL       ;;SET THE ZERO FILL SWITCH
025402  112737  000006  025575          MOVB    #6,$OMODE+1     ;;SET FOR SIX(6) DIGITS
025410  112737  000005  025572  $TYPON: MOVB    #5,$OCNT        ;;SET THE ITERATION COUNT
025416  010346                  MOV     R3,-(SP)        ;;SAVE R3
025420  010446                  MOV     R4,-(SP)        ;;SAVE R4
025422  010546                  MOV     R5,-(SP)        ;;SAVE R5
025424  113704  025575          MOVB    $OMODE+1,R4     ;;GET THE NUMBER OF DIGITS TO TYPE
025430  005404                  NEG     R4
025432  062704  000006          ADD     #6,R4           ;;SUBTRACT IT FOR MAX. ALLOWED
025436  110437  025574          MOVB    R4,$OMODE       ;;SAVE IT FOR USE
025442  113704  025573          MOVB    $OFILL,R4       ;;GET THE ZERO FILL SWITCH
025446  016605  000012          MOV     12(SP),R5       ;;PICKUP THE INPUT NUMBER
025452  005003                  CLR     R3              ;;CLEAR THE OUTPUT WORD
025454  006105          1$:     ROL     R5              ;;ROTATE MSB INTO "C"
025456  000404                  BR      3$              ;;GO DO MSB
025460  006105          2$:     ROL     R5              ;;FORM THIS DIGIT
025462  006105                  ROL     R5
025464  006105                  ROL     R5
025466  010503                  MOV     R5,R3
025470  006103          3$:     ROL     R3              ;;GET LSB OF THIS DIGIT
025472  105337  025574          DECB    $OMODE          ;;TYPE THIS DIGIT?
025476  100016                  BPL     7$              ;;BR IF NO
025500  042703  177770          BIC     #177770,R3      ;;GET RID OF JUNK
025504  001002                  BNE     4$              ;;TEST FOR 0
025506  005704                  TST     R4              ;;SUPPRESS THIS 0?
025510  001403                  BEQ     5$              ;;BR IF YES
```

F11


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

```
                                .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.

025576  010046                  $TRAP:  MOV     R0,-(SP)        ;;SAVE R0
025600  016600  000002                  MOV     2(SP),R0        ;;GET TRAP ADDRESS
025604  005740                          TST     -(R0)           ;;BACKUP BY 2
025606  111000                          MOVB    (R0),R0         ;;GET RIGHT BYTE OF TRAP
025610  006300                          ASL     R0              ;;POSITION FOR INDEXING
025612  016000  025632                  MOV     $TRPAD(R0),R0   ;;INDEX TO TABLE
025616  000200                          RTS     R0              ;;GO TO ROUTINE

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

025620  011646                  $TRAP2: MOV     (SP),-(SP)      ;;MOVE THE PC DOWN
025622  016666  000004  000002          MOV     4(SP),2(SP)     ;;MOVE THE PSW DOWN
025630  000002                          RTI                     ;;RESTORE THE PSW

                                .SBTTL  TRAP TABLE

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

                                ;       ROUTINE
                                ;       -------
025632  025620                  $TRPAD: .WORD   $TRAP2
025634  024620                          $TYPE   ;;CALL=TYPE     TRAP+1(104401)  TTY TYPEOUT ROUTINE
025636  025374                          $TYPOC  ;;CALL=TYPOC    TRAP+2(104402)  TYPE OCTAL NUMBER (WITH LEADING ZEROS)
025640  025350                          $TYPOS  ;;CALL=TYPOS    TRAP+3(104403)  TYPE OCTAL NUMBER (NO LEADING ZEROS)
025642  025410                          $TYPON  ;;CALL=TYPON    TRAP+4(104404)  TYPE OCTAL NUMBER (AS PER LAST CALL)
```

```
                                        .SBTTL  POWER DOWN AND UP ROUTINES
                                        ;;**********************************************************************
                                        ;POWER DOWN ROUTINE
025644  012737  026016  000024  $PWRDN: MOV     #$ILLUP,@#PWRVEC ;;SET FOR FAST UP
025652  012737  000340  000026          MOV     #340,@#PWRVEC+2  ;;PRIO:7
025660  010046                          MOV     R0,-(SP)         ;;PUSH R0 ON STACK
025662  010146                          MOV     R1,-(SP)         ;;PUSH R1 ON STACK
025664  010246                          MOV     R2,-(SP)         ;;PUSH R2 ON STACK
025666  010346                          MOV     R3,-(SP)         ;;PUSH R3 ON STACK
025670  010446                          MOV     R4,-(SP)         ;;PUSH R4 ON STACK
025672  010546                          MOV     R5,-(SP)         ;;PUSH R5 ON STACK
025674  017746  153244                  MOV     @SWR,-(SP)       ;;PUSH @SWR ON STACK
025700  010637  026022                  MOV     SP,$SAVR6        ;;SAVE SP
025704  012737  025716  000024          MOV     #$PWRUP,@#PWRVEC ;;SET UP VECTOR
025712  000000                          HALT
025714  000776                          BR      .-2              ;;HANG UP
                                        ;;**********************************************************************
                                        ;POWER UP ROUTINE
025716  012737  026016  000024  $PWRUP: MOV     #$ILLUP,@#PWRVEC ;;SET FOR FAST DOWN
025724  013706  026022                  MOV     $SAVR6,SP        ;;GET SP
025730  005037  026022                  CLR     $SAVR6           ;;WAIT LOOP FOR THE TTY
025734  005237  026022          1$:     INC     $SAVR6           ;;WAIT FOR THE INC
025740  001375                          BNE     1$               ;;OF  WORD
025742  011600                          MOV     (SP),R0          ;GET SAVED SWR OFF STACK
025744  076600  000226                  MED     ,226             ;RESTORE SWR CONTENTS
025750  012677  153170                  MOV     (SP)+,@SWR       ;;POP STACK INTO @SWR
025754  012605                          MOV     (SP)+,R5         ;;POP STACK INTO R5
025756  012604                          MOV     (SP)+,R4         ;;POP STACK INTO R4
025760  012603                          MOV     (SP)+,R3         ;;POP STACK INTO R3
025762  012602                          MOV     (SP)+,R2         ;;POP STACK INTO R2
025764  012601                          MOV     (SP)+,R1         ;;POP STACK INTO R1
025766  012600                          MOV     (SP)+,R0         ;;POP STACK INTO R0
025770  012737  025644  000024          MOV     #$PWRDN,@#PWRVEC ;;SET UP THE POWER DOWN VECTOR
025776  012737  000340  000026          MOV     #340,@#PWRVEC+2  ;;PRIO:7
026004  104401                          TYPE                     ;REPORT THE POWER FAILURE
026006  026024          $PWRMG: .WORD   $POWER                   ;;POWER FAIL MESSAGE POINTER
026010  012716                  MOV     (PC)+,(SP)               ;;RESTART AT START
026012  003000          $PWRAD: .WORD   START                    ;;RESTART ADDRESS
026014  000002                  RTI
026016  000000          $ILLUP: HALT                             ;;THE POWER UP SEQUENCE WAS STARTED
026020  000776                  BR      .-2                      ;; BEFORE THE POWER DOWN WAS COMPLETE
026022  000000          $SAVR6: 0                                ;;PUT THE SP HERE
026024  005015  047520  042527  $POWER: .ASCIZ  <15><12>"POWER"
026032  000122
                                        .EVEN
```

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

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

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

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

```
5623  030313     120  020103  047515  EMAA:   .ASCIZ  "PC MODE 2 ADDR - INCRE OF +0"
5624  030320  042504  031040  040440
5625  030326  042104  020122  020055
5626  030334  047111  051103  020105
5627  030342  043117  025440  000060
5628  030350  041520  046440  042117  EMAB:   .ASCIZ  "PC MODE 2 ADDR - INCRE OF +4, +6, OR +10"
5629  030356  020105  020062  042101
5630  030364  051104  026440  044440
5631  030372  041516  042522  047440
5632  030400  020106  032053  020054
5633  030406  033053  020054  051117
5634  030414  025440  030061     000
5635  030421     120  020103  047515  EMAC:   .ASCIZ  "PC MODE 2 ADDR - INCRE OF +6 OR +10"
5636  030426  042504  031040  040440
5637  030434  042104  020122  020055
5638  030442  047111  051103  020105
5639  030450  043117  025440  020066
5640  030456  051117  025440  030061
5641  030464     000
5642  030465     120  020103  047515  EMAD:   .ASCIZ  "PC MODE 2 ADDR - INCRE OF +10"
5643  030472  042504  031040  040440
5644  030500  042104  020122  020055
5645  030506  047111  051103  020105
5646  030514  043117  025440  030061
5647  030522     000
5648
5649
```

```
                                                    ; DATA HEADERS HERE
030523    114  040517  042504  DHA:    .ASCIZ  "LOADED STORED"
030530 004504  052123  051117
030536 042105     000
030541    123  047524  042522  DHB:    .ASCIZ  "STORED"
030546 000104
030550 026455  026455  047514  DHC:    .ASCIZ  "----LOADED---- ----STORED----"
030556 042101  042105  026455
030564 026455  026411  026455
030572 051455  047524  042522
030600 026504  026455  000055
030606 054105  023520  004504  DHD:    .ASCIZ  "EXP'D  RCV'D"
030614 041522  023526  000104
030622 054105  023520  026504  DHE:    .ASCIZ  "EXP'D-FEC-RCV'D        EXP'D-FEA-RCV'D"
030630 042506  026503  041522
030636 023526  004504  054105
030644 023520  026504  042506
030652 026501  041522  023526
030660 000104
030662 026455  042455  050130  DHF:    .ASCIZ  "---EXPECTED--- ---RECEIVED---"
030670 041505  042524  026504
030676 026455  026411  026455
030704 042522  042503  053111
030712 042105  026455  000055
030720 026455  026455  026455  DHG:    .ASCII  "----------EXPECTED---------- "
030726 026455  026455  042455
030734 050130  041505  042524
030742 026504  026455  026455
030750 026455  026455  026455
030756     011
030757     055  026455  026455          .ASCIZ  "----------RECEIVED----------"
030764 026455  026455  026455
030772 042522  042503  053111
031000 042105  026455  026455
031006 026455  026455  026455
031014 000055
031016 046117  020104  041520  DHH:    .ASCIZ  "OLD PC OLD PS  NEW SP   FPS     FEC     FEA"
031024 047411  042114  050040
031032 004523  042516  020127
031040 050123  020011  050106
031046 004523  043040  041505
031054 020011  042506  000101
031062 047514  042101  042105  DHI:    .ASCIZ  "LOADED STORED    R2"
031070 051411  047524  042522
031076 004504  020040  031122
031104     000
031105    114  040517  042504  DHJ:    .ASCIZ  "LOADED STORED    R3"
031112 004504  052123  051117
031120 042105  020011  051040
031126 000063
031130 047514  042101  042105  DHK:    .ASCIZ  "LOADED STORED    R0      R2"
031136 051411  047524  042522
031144 004504  020040  030122
031152 020011  051040  000062
031160 047514  042101  042105  DHL:    .ASCIZ  "LOADED STORED  $REG2     R0"
031166 051411  047524  042522
```

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

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

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

```
5874  033776  026455  026455  047514  DHAJ:   .ASCII  "----LOADED---- ----STORED----  $REG4"
5875  034004  042101  042105  026455
5876  034012  026455  026411  026455
5877  034020  051455  047524  042522
5878  034026  026504  026455  004455
5879  034034  051044  043505     064
5880  034041     011  051044  043505          .ASCIZ  "       $REG5     R0      R2"
5881  034046  004465  020040  030122
5882  034054  020011  051040  000062
5883  034062  026455  026455  047514  DHAK:   .ASCIZ  "----LOADED---- ----STORED----"
5884  034070  042101  042105  026455
5885  034076  026455  026411  026455
5886  034104  051455  047524  042522
5887  034112  026504  026455  000055
5888  034120  000040                  DHAL:   .ASCIZ  " "
5889
5890
```

```
5891                                                    ; DATA ADDRESS VECTOR
5892                                                    .EVEN
5893  033122  001170  000000          DTA:    .WORD   $REG0,0
5894  033126  001170  001172  000000  DTB:    .WORD   $REG0,$REG1,0
5895  033134  001170  001172  001174  DTC:    .WORD   $REG0,$REG1,$REG2,$REG3,0
5896  033142  001176  000000
5897  033146  001242  002330  000000  DTD:    .WORD   $TMPS,FPS,0
5898  033154  001252  002330  000000  DTE:    .WORD   $TMP11,FPS,0
5899  033162  001256  002330  000000  DTF:    .WORD   $TMP13,FPS,0
5900  033170  001244  002332  002344  DTG:    .WORD   $TMP6,FEC,EXPFEA,FEA,0
5901  033176  002334  000000
5902  033202  001254  002332  002344  DTH:    .WORD   $TMP12,FEC,EXPFEA,FEA,0
5903  033210  202334  000000
5904  033214  001260  002332  002344  DTI:    .WORD   $TMP14,FEC,EXPFEA,FEA,0
5905  033222  002334  000000
5906  033226  001234  001236  001240  DTJ:    .WORD   $TMP2,$TMP3,$REG0,$REG1,0
5907  033234  001172  000000
5908  033240  001240  001242  001244  DTK:    .WORD   $TMP4,$TMP5,$TMP6,$TMP7
5909  033246  001246
5910  033250  001170  001172  001174          .WORD   $REG0,$REG1,$REG2,$REG3,0
5911  033256  001176  000000
5912  033262  001244  001246  001250  DTL:    .WORD   $TMP6,$TMP7,$TMP10,$TMP11
5913  033270  001252
5914  033272  001170  001172  001174          .WORD   $REG0,$REG1,$REG2,$REG3,0
5915  033300  001176  000000
5916  033304  002336  002340  002342  DTM:    .WORD   FPPOPC,FPPOPS,FPPOSP,FPS,FEC,FEA,0
5917  033312  002330  002332  002334
5918  033320  000000
5919  033322  001170  001172  002352  DTN:    .WORD   $REG0,$REG1,EREG2,0
5920  033330  000000
5921  033332  001170  001172  002354  DTO:    .WORD   $REG0,$REG1,EREG3,0
5922  033340  000000
5923  033342  001170  001172  002346  DTP:    .WORD   $REG0,$REG1,EREG0,EREG2,0
5924  033350  002352  000000
5925  033354  001170  001172  001174  DTQ:    .WORD   $REG0,$REG1,$REG2,EREG0,0
5926  033362  002346  000000
5927  033366  001170  001172  001174  DTR:    .WORD   $REG0,$REG1,$REG2,EREG2,0
5928  033374  002352  000000
5929  033400  001170  001172  001174  DTS:    .WORD   $REG0,$REG1,$REG2,EREG4,0
5930  033406  002356  000000
5931  033412  001170  001172  001174  DTT:    .WORD   $REG0,$REG1,$REG2,EREG5,0
5932  033420  002360  000000
5933  033424  001170  001172  001174  DTU:    .WORD   $REG0,$REG1,$REG2,EREG1,EREG3,0
5934  033432  002350  002354  000000
5935  033440  001170  001172  001174  DTV:    .WORD   $REG0,$REG1,$REG2,EREG3,EREG5,0
5936  033446  002354  002360  000000
5937  033454  001200  001202  001204  DTW:    .WORD   $REG4,$REG5,$REG6,$REG7,0
5938  033462  001206  000000
5939  033466  001200  001202  001204  DTX:    .WORD   $REG4,$REG5,$REG6,$REG7,EREG4,0
5940  033474  001206  002356  000000
5941  033502  001200  001202  001204  DTY:    .WORD   $REG4,$REG5,$REG6,$REG7,EREG5,0
5942  033510  001206  002360  000000
5943  033516  001200  001202  001204  DTZ:    .WORD   $REG4,$REG5,$REG6,$REG7,EREG0,EREG3,0
5944  033524  001206  002346  002354
5945  033532  000000
5946  033534  001200  001202  001204  DTAA:   .WORD   $REG4,$REG5,$REG6,$REG7,EREG1,EREG3,0
```

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

```
ABASE = 000000            325
ABSDI   022140           4594#   4604
ABSDL   022104           4585    4587#
ABSDT   022060           3075    3094    3113    3132    3151    3170    3189    3208    3227    3246    3265    4579#
ABSD1   014640           3074    3079#
ABSD10  015216           3207    3212#
ABSD11  015260           3226    3231#
ABSD12  015322           3245    3250#
ABSD13  015364           3264    3269#
ABSD2   014702           3093    3098#
ABSD3   014744           3112    3117#
ABSD4   015006           3131    3136#
ABSD5   015050           3150    3155#
ABSD6   015112           3169    3174#
ABSD7   015154           3188    3193#
ABSFI   021750           4549#   4559
ABSFL   021730           4542    4544#
ABSFT   021704           2887    2904    2921    2938    2955    2972    2989    3006    3023    3040    3057    4536#
ABSF1   014202           2886    2891#
ABSF10  014470           3005    3010#
ABSF11  014522           3022    3027#
ABSF12  014554           3039    3044#
ABSF13  014606           3056    3061#
ABSF2   014234           2903    2908#
ABSF3   014266           2920    2925#
ABSF4   014320           2937    2942#
ABSF5   014352           2954    2959#
ABSF6   014404           2971    2976#
ABSF7   014436           2988    2993#
ACDW1 = 000000            325
ACDW2 = 000000            325
ACPUOP= 000000            325     340
ADDR1   004142            744     759#
ADDR10  004670            931     953#
ADDR11  004756            960     985#
ADDR12  005020            992    1006#
ADDR13  005060           1013    1027#
ADDR14  005136           1034    1054#
ADDR2   004222            766     787#
ADDR3   004312            794     820#
ADDR4   004370            827     847#
ADDR5   004442            854     874#
ADDR6   004520            881     901#
ADDR7   004574            908     924#
ADDW0 = 000000            325
ADDW1 = 000000            325
ADDW10= 000000            325
ADDW11= 000000            325
ADDW12= 000000            325
ADDW13= 000000            325
ADDW14= 000000            325
ADDW15= 000000            325
ADDW2 = 000000            325
ADDW3 = 000000            325
ADDW4 = 000000            325
ADDW5 = 000000            325
```

```
A0046  = 000000            325
A0047  = 000000            325
A0048  = 000000            325
A0049  = 000000            325
AEVCT= 000000              325     331
AEVM   = 000000            325
ADR01   010146            2009    2010    2011    2012    2032#
ADR010  011426            2271    2272    2273    2274    2295#
ADR011  011564            2304    2305    2306    2307    2334#
ADR012  011724            2343    2344    2345    2346    2372#
ADR013  012050            2381    2382    2383    2384    2404#
ADR014  012214            2413    2414    2415    2416    2443#
ADR02   010302            2041    2042    2043    2044    2069#
ADR03   010456            2078    2079    2080    2081    2109#
ADR04   010612            2118    2119    2120    2121    2146#
ADR05   010750            2155    2156    2157    2158    2184#
AJR06   011120            2193    2194    2195    2196    2223#
ADR07   011272            2232    2233    2234    2235    2262#
ADRF1   012316            2452    2453    2475#
ADRF10  013252            2667    2668    2696#
ADRF11  013360            2704    2705    2727#
ADRF12  013442            2735    2736    2751#
ADRF13  013542            2759    2760    2781#
ADRF14  013632            2789    2790    2807#
ADRF2   012424            2483    2484    2507#
ADRF3   012546            2515    2516    2545#
ADRF4   012644            2553    2554    2572#
ADRF5   012740            2580    2581    2602#
ADRF6   013042            2610    2611    2629#
ADRF7   013136            2637    2638    2659#
AENV   = 000000            325     336
AENVM  = 000000            325     337
AFATAL= 000000             325     328
ALTN   = 125252            163#    481     1157    1159    1195    1593    1594    1610    613     1648    1651    2109    2184
                           2404    2443    2572    2629    2926    2927    3118    3120    3324    3325    3516    3518    3739
                           3848    3940    3941    4132    4134
ALTP   = 052525            162#    481     1139    1141    1195    1591    1594    1612    1613    1648    1651    2032    2334
                           2507    2696    2751    2909    2910    3099    3101    3307    3308    3497    3499    3722    3829
                           3923    3924    4113    4115
ALT4N  = 170360            165#    2069    2295    2372    2475    2659
ALT4P  = 007417            164#    2146    2223    2545    2602    2781    2807
AMADR1= 000000             325
AMADR2= 000000             325
AMADR3= 000000             325
AMADR4= 000000             325
AMAMS1= 000000             325
AMAMS2= 000000             325
AMAMS3= 000000             325
AMAMS4= 000000             325
AMSGAD= 000000             325     333
AMSGLG= 000000             325     334
AMSGTY= 000000             325     327
AMTYP1= 000000             325
AMTYP2= 000000             325
AMTYP3= 000000             325
AMTYP4= 000000             325
```

```
APASS = 000000            325     330
APRIOR= 000000            325
APTCSU= 000040           5132    5237#
APTENV= 000001           5012    5125    5193    5235#
APTSIZ= 000200            548    5234#
APTSPO= 000100           5127    5195    5236#
ASCHOT  026034            631    5402#
ASCWRM  026042            633    5403#
ASWREG= 000000            325     338
ATESTN= 000000            325     329
AUNIT = 000000            325     332
AUSWR = 000000            325     339
AVECT1= 000000            325
AVECT2= 000000            325
BGNMES  002376            486#    557
BIT0  = 000001            126#
BIT00 = 000001            116#    126     639     661
BIT01 = 000002            115#    125
BIT02 = 000004            114#    124
BIT03 = 000010            113#    123
BIT04 = 000020            112#    122     563     613     643     665    4313
BIT05 = 000040            111#    121    1073
BIT06 = 000100            110#    120     714     730
BIT07 = 000200            109#    119     682     698
BIT08 = 000400            108#    118    4939
BIT09 = 001000            107#    117    4947    5022
BIT1  = 000002            125#
BIT10 = 002000            106#   5000
BIT11 = 004000            105#   4954
BIT12 = 010000            104#    595     623     625     628     645     667    4319
BIT13 = 020000            103#    645     667    5007
BIT14 = 040000            102#   4925
BIT15 = 100000            101#
BIT2  = 000004            124#
BIT3  = 000010            123#
BIT4  = 000020            122#
BIT5  = 000040            121#
BIT6  = 000100            120#
BIT7  = 000200            119#
BIT8  = 000400            118#
BIT9  = 001000            117#
BPTVEC= 000014            133#
CCONLY= 177760            182#   1071    1080
CLRDI   023110           4776#   4786
CLRDL   023054           4767    4769#
CLRDT   023030           3786    3805    3824    3843    3862    3881    4761#
CLRD1   017106           3785    3[illegible]#
CLRD2   017150           3804    3809#
CLRD3   017212           3823    3828#
CLRD4   017254           3842    3847#
CLRD5   017316           3861    3866#
CLRD6   017360           3880    3885#
CLRFI   022720           4731#   4741
CLRFL   022700           4724    4726#
CLRFT   022654           3683    3700    3717    3734    3751    3768    4718#
CLRF1   016652           3682    3687#
```

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

```
DTAG    033666  431  5966#
DTAH    033702  422  423  425  5968#
DTAI    033716  414  415  5970#
DTAJ    033734  429  5973#
DTAK    033750  416  417  5975#
DTAL    033770  421  5979#
DTAM    034010  426  427  428  5983#
DTAN    034030  424  5987#
DTAO    034050  418  419  420  5991#
DTAP    034072  456  457  458  459  5995#
DTB     033126  368  369  393  5894#
DTC     033134  430  433  5895#
DTD     033146  435  5897#
DTE     033154  436  5898#
DTF     033162  437  5899#
DTG     033170  439  5900#
DTH     033202  440  5902#
DTI     033214  441  5904#
DTJ     033226  445  447  449  451  5906#
DTK     033240  443  446  448  450  452  5908#
DTL     033262  444  5912#
DTM     033304  454  5916#
DTN     033322  390  5919#
DTO     033332  388  389  5921#
DTP     033342  383  384  5923#
DTQ     033354  381  382  5925#
DTR     033366  387  5927#
DTS     033400  385  386  391  5929#
DTT     033412  375  5931#
DTU     033424  378  379  380  5933#
DTV     033440  376  377  5935#
DTW     033454  411  5937#
DTX     033466  396  397  5939#
DTY     033502  395  400  5941#
DTZ     033516  401  402  5943#
EMA     026051  435  436  437  5407#
EMAA    030313  456  5623#
EMAB    030350  457  5628#
EMAC    030421  458  5635#
EMAD    030465  459  5642#
EMB     026075  439  440  441  5411#
EMC     026125  445  446  5416#
EMD     026173  454  5423#
EME     026260  433  5432#
EMF     026331  368  5439#
EMG     026366  369  5444#
EMH     026423  370  5449#
EMI     026462  371  5455#
EMJ     026517  372  5460#
EMK     026556  373  5466#
EML     026613  375  377  380  382  384  386  387  389  390  5471#
EMM     026703  378  381  383  385  5481#
EMN     027005  376  379  388  391  5493#
EMO     027107  393  5505#
EMP     027165  395  397  399  402  404  405  406  408  410  411  5513#
EMQ     027261  396  401  403  407  5524#
```

```
EMR      027370            398     400     409     412    5537#
EMS      027477            415     417     420     421     423     424     428     429     430     431    5550#
EMT      027573            416     418     425     426    5561#
EMTVEC=  000030            136#    519*    520*
EMU      027702            414     419     422     427    5574#
EMV      030011            443    5587*
EMV001   001354            368#
EMV002   001362            369#
EMV003   001370            370#
EMV004   001376            371#
EMV005   001404            372#
EMV006   001412            373#
EMV007   001420            375#
EMV010   001426            376#
EMV011   001434            377#
EMV012   001442            378#
EMV013   001450            379#
EMV014   001456            380#
EMV015   001464            381#
EMV016   001472            382#
EMV017   001500            383#
EMV020   001506            384#
EMV021   001514            385#
EMV022   001522            386#
EMV023   001530            387#
EMV024   001536            388#
EMV025   001544            389#
EMV026   001552            390#
EMV027   001560            391#
EMV030   001566            393#
EMV031   001574            395#
EMV032   001602            396#
EMV033   001610            397#
EMV034   001616            398#
EMV035   001624            399#
EMV036   001632            400#
EMV037   001640            401#
EMV040   001646            402#
EMV041   001654            403#
EMV042   001662            404#
EMV043   001670            405#
EMV044   001676            406#
EMV045   001704            407#
EMV046   001712            408#
EMV047   001720            409#
EMV050   001726            410#
EMV051   001734            411#
EMV052   001742            412#
EMV053   001750            414#
EMV054   001756            415#
EMV055   001764            416#
EMV056   001772            417#
EMV057   002000            418#
EMV060   002006            419#
EMV061   002014            420#
EMV062   002022            421#
```

```
EMV063  002030            422#
EMV064  002036            423#
EMV065  002044            424#
EMV066  002052            425#
EMV067  002060            426#
EMV070  002066            427#
EMV071  002074            428#
EMV072  002102            429#
EMV073  002110            430#
EMV074  002116            431#
EMV075  002124            433#
EMV076  002132            435#
EMV077  002140            436#
EMV100  002146            437#
EMV101  002154            439#
EMV102  002162            440#
EMV103  002170            441#
EMV104  002176            443#
EMV105  002204            444#
EMV106  002212            445#
EMV107  002220            446#
EMV110  002226            447#
EMV111  002234            448#
EMV112  002242            449#
EMV113  002250            450#
EMV114  002256            451#
EMV115  002264            452#
EMV116  002272            454#
EMV117  002300            456#
EMV120  002306            457#
EMV121  002314            458#
EMV122  002322            459#
EMW     030047    444     5593#
EMX     030111    447     448     5599#
EMY     030157    451     452     5606#
EMZ     030245    449     450     5616#
EREG0   002346    471#    4988#   5923    5925    5943    5949    5977    5993
EREG1   002350    472#    4989#   5933    5946    5960    5966    5981
EREG2   002352    473#    4990#   5919    5923    5927    5952    5970    5981    5993
EREG3   002354    474#    4991#   5921    5933    5935    5943    5946    5968
EREG4   002356    475#    4992#   5929    5939    5955    5964    5977    5985
EREG5   002360    476#    4993#   5931    5935    5941    5964    5970    5985    5989
EREG6   002362    477#    4994#   4995#
EREG7   002364    478#    4996#
ERROR1= 104117    185#    937     985     2262    2313    2727    2765
ERRVEC= 000004    129#    533     534#    545#    4930    4931#   4933#   4936#
EXPFEA  002344    468#    4393#   4394#   4397    4468#   4469#   4472    4559#   4562    4604#   4607    4650#   4653
                  4695#   4698    4741#   4744    4786#   4789    4833#   4836    4878#   4881    5900    5902    5904
FEA     002334    464#    4397    4472    4562    4607    4653    4698    4744    4789    4836    4881    5900    5902
                  5904    5916
FEC     002332    463#    4386#   4395    4461#   4470    4552#   4560    4597#   4605    4643#   4651    4688#   4696
                  4734#   4742    4779#   4787    4826#   4834    4871#   4879    4905#   5900    5902    5904    5916
FPPILT  023624    554     4901#
FPPOPC  002336    465#    4902#   4908    5916
FPPOPS  002340    466#    4903#   4907    5916
FPPOSP  002342    467#    4901#   5916
```

```
FPPVEC= 000244          151#   554#   555#
FPS     002330          462#  4385#  4388  4460#  4463  4551#  4554  4596#  4599  4642#  4645  4687#  4690
                       4733#  4736  4778#  4781  4825#  4828  4870#  4873  4904#  5897  5898  5899  5916
FPS0  = 000000          179#   695   698   727   730
FPS1  = 147757          178#   679   682   711   714
FIN   = 140200          175#
FIP   = 040200          174#
GNS   = ****** U        193   5346  5347  5348  5349
HOTWRM  024406          631#   633#  5050#
HT    = 000011           39#  5140  5181
IOTVEC= 000020          134#   517#   518#
LDAOL   021056         4379  4381#
LDAOT   021032         1098  1116  1134  1152  1170  1188  1206  1218  1230  1242  1254  1266  1279
                       1291  1303  1315  1327  1339  1352  1364  1376  1388  1400  1412  1425  1437
                       1449  1461  1473  1486  1498  1510  1522  1534  4373#
LDAO1   005312         1097  1102#  1205  1278  1351  1424  1485
LDAO2   005360         1115  1120#  1217  1290  1363  1436  1497
LDAO3   005426         1133  1138#  1229  1302  1375  1448  1509
LDAO4   005474         1151  1156#  1241  1314  1387  1460  1521
LDAO5   005542         1169  1174#  1253  1326  1399  1472  1533
LDAO6   005610         1187  1192#  1265  1338  1411
LDAFL   021360         4454  4456#
LDAFT   021334         1548  1567   1586  1605  1624  1643  1663  1675  1687  1699  1711  1723  1736
                       1748  1760   1772  1784  1796  1809  1821  1833  1845  1857  1869  1882  1894
                       1906  1918   1930  1943  1955  1967  1979  1991  4448#
LDAF1   006556         1547  1552#  1662  1735  1808  1881  1942
LDAF2   006630         1566  1571#  1674  1747  1820  1893  1954
LDAF3   006702         1585  1590#  1686  1759  1832  1905  1966
LDAF4   006754         1604  1609#  1698  1771  1844  1917  1978
LDAF5   007026         1623  1628#  1710  1783  1856  1929  1990
LDAF6   007100         1642  1647#  1722  1795  1868
LDAR00  021214         1096  1114   1132  1150  1168  1186  4417#
LDAR01  021230         1204  1216   1228  1240  1252  1264  4421#
LDAR02  021244         1277  1289   1301  1313  1325  1337  4425#
LDAR03  021260         1350  1362   1374  1386  1398  1410  4429#
LDAR04  021274         1423  1435   1447  1459  1471  4433#
LDAR05  021314         1484  1496   1508  1520  1532  4439#
LDARF0  021514         1546  1565   1584  1603  1622  1641  4492#
LDARF1  021536         1661  1673   1685  1697  1709  1721  4498#
LDARF2  021560         1734  1746   1758  1770  1782  1794  4504#
LDARF3  021602         1807  1819   1831  1843  1855  1867  4510#
LDARF4  021624         1880  1892   1904  1916  1928  4516#
LDARF5  021654         1941  1953   1965  1977  1989  4525#
LF    = 000012           40#   486   491   499  5175  5181
LGN   = 177777          169#  2943  3137  3341  3376  3535  3575  3957  3958  4151  4153
LGP   = 077777          168#  2944  2977  2978  3139  3175  3177  3342  3375  3537  3573  3991  3992
                       4189  4191
MED   = 076600          143#   562   612   616   626  4312  4320  4326  5378
MO    = 100000          167#  1193  1631  1632  1650  3028  3045  3232  3251  3426  3443  3630  3649
                       3756  3867  4042  4043  4059  4060  4246  4248  4265  4267
M1    = 177777          166#   481  1121  1123  1175  1177  1195  1555  1556  1572  1575  1629  1632
                       1648  1651  2943  2944  2977  2978  3011  3062  3137  3139  3175  3177  3213
                       3270  3341  3342  3375  3376  3409  3460  3535  3537  3573  3575  3611  3668
                       3705  3773  3810  3886  3957  3958  3991  3992  4025  4026  4076  4077  4151
                       4153  4189  4191  4227  4229  4284  4286
NEGDI   022524         4685#  4695
```

```
NEGDL   022470          4676    4678#
NEGDT   022444          3473    3492    3511    3530    3549    3568    3587    3606    3625    3644    3663    4670#
NEGD1   016064          3472    3477#
NEGD10  016442          3605    3610#
NEGD11  016504          3624    3629#
NEGD12  016546          3643    3648#
NEGD13  016610          3662    3667#
NEGD2   016126          3491    3496#
NEGD3   016170          3510    3515#
NEGD4   016232          3529    3534#
NEGD5   016274          3548    3553#
NEGD6   016336          3567    3572#
NEGD7   016400          3586    3591#
NEGFI   022334          4640#   4650
NEGFL   022314          4633    4635#
NEGFT   022270          3285    3302    3319    3336    3353    3370    3387    3404    3421    3438    3455    4627#
NEGF1   015426          3284    3289#
NEGF10  015714          3403    3408#
NEGF11  015746          3420    3425#
NEGF12  016000          3437    3442#
NEGF13  016032          3454    3459#
NEGF2   015460          3301    3306#
NEGF3   015512          3318    3323#
NEGF4   015544          3335    3340#
NEGF5   015576          3352    3357#
NEGF6   015630          3369    3374#
NEGF7   015662          3386    3391#
NEMPAS  003434           566     568    593#    4368
NMPAS1  002475           499#    598
OFCC    005252          1067    1088#
PIRQ    = 177772          46#
PIRQVE= 000240           140#
PREVAC  002366           481#   4417    4421    4425    4429    4433    4439
PR0     = 000000          63#
PR1     = 000040          64#
PR2     = 000100          65#
PR3     = 000140          66#
PR4     = 000200          67#
PR5     = 000240          68#
PR6     = 000300          69#
PR7     = 000340          70#
PS      = 177776          43#     44
PSW     = 177776          44#   1079
PWRVEC= 000024           135#    523*    524*   5356*   5357*   5366*   5372*   5386*   5387*
RESVEC= 000010           130#
RFLAG   = 000144           147#    616    4320
RMMAMI= 000022           145#    562     612    4312
SMN     = 100200           171#   2994    3194    3359    3392    3556    3592    4008    4009    4208    4210
SMP     = 000200           170#   2960    2961    2995    3156    3158    3196    3358    3393    3554    3594    3974    3975
                          4170   4172
STACK   = 001100            34#    515     593     610
START   003000           197     508#   5391
STKLMT= 177774            45#
SUBPAS  003472           596     610#   4330
SWR     001144           262#    513     535*    537     543#    550#    595     618     621    4316    4355    4925    4939
                          4947   4954    5000    5007    5019    5022    5364    5379*
```

```
SWREG   000176          195#    543
SW0     = 000001         98#
SW00    = 000001         88#     98     621
SW01    = 000002         87#     97     618    4316
SW02    = 000004         86#     96
SW03    = 000010         85#     95
SW04    = 000020         84#     94
SW05    = 000040         83#     93
SW06    = 000100         82#     92
SW07    = 000200         81#     91
SW08    = 000400         80#     90
SW09    = 001000         79#     89
SW1     = 000002         97#
SW10    = 002000         78#   4355
SW11    = 004000         77#
SW12    = 010000         76#
SW13    = 020000         75#
SW14    = 040000         74#
SW15    = 100000         73#
SW2     = 000004         96#
SW3     = 000010         95#
SW4     = 000020         94#
SW5     = 000040         93#
SW6     = 000100         92#
SW7     = 000200         91#
SW8     = 000400         90#
SW9     = 001000         89#
TBITVE= 000014          131#
TKVEC   = 000060        138#
TPVEC   = 000064        139#
TRAPVE= 000034          137#    521#    522#
TRTVEC= 000014          132#
TSTDI   023474         4868#   4878
TSTDL   023440         4859    4861#
TSTDT   023414         4089    4108   4127   4146   4165   4184   4203   4222   4241   4260   4279   4853#
TSTD1   020060         4088    4093#
TSTD10  020436         4221    4226#
TSTD11  020500         4240    4245#
TSTD12  020542         4259    4264#
TSTD13  020604         4278    4283#
TSTD2   020122         4107    4112#
TSTD3   020164         4126    4131#
TSTD4   020226         4145    4150#
TSTD5   020270         4164    4169#
TSTD6   020332         4183    4188#
TSTD7   020374         4202    4207#
TSTFI   023304         4823#   4833
TSTFL   023264         4816    4818#
TSTFT   023240         3901    3918   3935   3952   3969   3986   4003   4020   4037   4054   4071   4810#
TSTF1   017422         3900    3905#
TSTF10  017710         4019    4024#
TSTF11  017742         4036    4041#
TSTF12  017774         4053    4058#
TSTF13  020026         4070    4075#
TSTF2   017454         3917    3922#
TSTF3   017506         3934    3939#
```

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

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

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

```
TST5    004022          710#
TST50   006176         1366    1373#
TST51   006216         1378    1385#
TST52   006236         1390    1397#
TST53   006256         1402    1409#
TST54   006276         1414    1422#
TST55   006316         1427    1434#
TST56   006336         1439    1446#
TST57   006356         1451    1458#
TST6    004050          726#
TST60   006376         1463    1470#
TST61   006416         1475    1483#
TST62   006436         1488    1495#
TST63   006456         1500    1507#
TST64   006476         1512    1519#
TST65   006516         1524    1531#
TST66   006536         1536    1545#
TST67   006610         1550    1564#
TST7    004076          743#
TST70   006662         1569    1583#
TST71   006734         1588    1602#
TST72   007006         1607    1621#
TST73   007060         1626    1640#
TST74   007132         1645    1660#
TST75   007152         1655    1672#
TST76   007172         1677    1684#
TST77   007212         1689    1696#
TYPE  = 104401          557     565     567     598     603    4357    5002    5010    5049    5068    5070    5071    5074
                       5076    5079    5082    5085    5093    5145    5298    5346#   5388
TYPOC = 104402         5058    5078    5081    5086    5347#
TYPON = 104404         5349#
TYPOS = 104403          601    5348#
WFLAG = 000344          148#    626    4326
ZX1MN = 100177          173#   1175    1177    3062    3270    3460    3668    4076    4077    4284    4286
ZX1MP = 000177          172#   3011    3213    3409    3611    3773    3886    4025    4026    4227    4229
$APTHD  001000          219     225#
$ASTAT= ****** U       5215    5230
$ATYC   025126         5186    5188#
$ATY1   025102         5184#
$ATY3   025110         5130    5185#
$ATY4   025120         5015    5187#
$AUTOB  001140          258#
$BOROR  001126          253#
$BODAT  001132          255#
$BELL   001314          317#    4357    5002    5034
$CHARC  025076         5147#   5157#   5164    5173#   5178#
$CKSUM= ****** U       5352
$CMTAG  001100          240#    510     511     519     525     526     527
$CM1  = 000020          275#    276#    277#    278#    279#    280#    281#    282#    283#    284#    285#    286#    287#
                        288#    289#    290#    291#
$CM2  = 000040          275#    276#    277#    278#    279#    280#    281#    282#    283#    284#    285#    286#    287#
                        288#    289#    290#    291#
$CM3  = 000020          273#    275
$CM4  = 000030          291#    292#    293#    294#    295#    296#    297#    298#    299#    300#    301#    302#    303#
                        304#    305#    306#    307#    308#    309#    310#    311#    312#    313#    314#    315#
$CPUOP  001352          340#
```

```
$CRLF   001321          319#    603     5010    5034    5050    5070    5076    5093    5146    5181
$CTR  = 000006         1200#   1212#   1224#   1236#   1248#   1260#   1272#   1273#   1285#   1297#   1309#   1321#   1333#
                       1345#   1346#   1358#   1370#   1382#   1394#   1406#   1418#   1419#   1431#   1443#   1455#   1467#
                       1479#   1480#   1492#   1504#   1516#   1528#   1540#   1657#   1669#   1681#   1693#   1705#   1717#
                       1729#   1730#   1742#   1754#   1766#   1778#   1790#   1802#   1803#   1815#   1827#   1839#   1851#
                       1863#   1875#   1876#   1888#   1900#   1912#   1924#   1936#   1937#   1949#   1961#   1973#   1985#
                       1997#
$DEVCT  001334          331#
$DOAGN  021026         4351    4360    4368#
$ENDAD  021016          205    4363#   5029
$ENDCT  020766          525    4353#
$ENV    001344          336#   5012    5125    5193    5217
$ENVM   001345          337#    548    5127    5132    5195
$EOP    020730         4314    4317    4339    4343#
$EOPCT  020760          525*   4350#   4351
$ERFLG  001104          244#   4299*   4344*   4943    4945    4951*   4973    4997*   5034
$ERMAX  001120          250#    528*   4945    4968*   4973
$ERROR  024142          519    4987#
$ERRPC  001122          251#   5004*   5005*   5006    5034    5057    5096
$ERRTB  001354          351#   5065
$ERTTL  001114          248#   5003*   5034
$ESCAP  001312          316#    527*   4967*   5025    5027    5034
$ETABL  001344          335#
$ETEND  001354          231     347#
$FATAL  001326          328#   5221*
$FFLG   025346         5184*   5187*   5215    5224*   5232#
$FILLC  001164          271#   5150    5181
$FILLS  001163          270#   5181
$GDADR  001124          252#
$GDDAT  001130          254#
$GET42  021006         4356    4359#
$GTSWR= ****** U       5351
$HD   = 000000           12
$HIBTS  001000          226#
$ICNT   001106          245#   4958*   4959    4961*   4972
$ILLUP  026016         5356    5372    5393#
$INTAG  001141          259#
$ITEMB  001116          249#   5006*   5014    5034    5054
$LF     001322          320#   5034    5181
$LFLG   025345         5225*   5231#
$LPADR  001110          246#    529*   4949*   4965*   4970    4972
$LPERR  001112          247#    530*    640*    662*   1068*   4379*   4454*   4542*   4585*   4633*   4676*   4724*   4767*
                       4816*   4859*   4949    4966*   4972    5024
$LPTST  001150          264#   4941
$MAIL   001324          227     231     326#    547     4964    5012    5125
$MBADR  001002          227#
$MFLG   025344         5185*   5191    5226*   5230#
$MSGAD  001340          333#   5201*   5204
$MSGLG  001342          334#   5206*
$MSGTY  001324          327#   5199    5207*   5219    5223*
$MXCNT  024140         4962    4972#
$NULL   001162          269#   5152    5181
$NWTST= 000001          635#    657#    675#    691#    707#    723#    740#    762#    790#    823#    850#    877#    904#
                        927#    956#    988#   1009#   1030#   1062#   1092#   1110#   1128#   1146#   1164#   1182#   1200#
                       1212#   1224#   1236#   1248#   1260#   1273#   1285#   1297#   1309#   1321#   1333#   1346#   1358#
                       1370#   1382#   1394#   1406#   1419#   1431#   1443#   1455#   1467#   1480#   1492#   1504#   1516#
```

```
                    1528#  1542#  1561#  1580#  1599#  1618#  1637#  1657#  1669#  1681#  1693#  1705#  1717#
                    1730#  1742#  1754#  1766#  1778#  1790#  1803#  1815#  1827#  1839#  1851#  1863#  1876#
                    1888#  1900#  1912#  1924#  1937#  1949#  1961#  1973#  1985#  2004#  2036#  2073#  2113#
                    2150#  2188#  2227#  2266#  2299#  2338#  2376#  2408#  2447#  2478#  2510#  2548#  2575#
                    2605#  2632#  2662#  2699#  2730#  2754#  2784#  2815#  2881#  2883   2898#  2900   2915#
                    2917   2932#  2934   2949#  2951   2966#  2968   2983#  2985   3000#  3002   3017#  3019
                    3034#  3036   3051#  3053   3069#  3071   3088#  3090   3107#  3109   3126#  3128   3145#
                    3147   3164#  3166   3183#  3185   3202#  3204   3221#  3223   3240#  3242   3259#  3261
                    3279#  3281   3296#  3298   3313#  3315   3330#  3332   3347#  3349   3364#  3366   3381#
                    3383   3398#  3400   3415#  3417   3432#  3434   3449#  3451   3467#  3469   3486#  3488
                    3505#  3507   3524#  3526   3543#  3545   3562#  3564   3581#  3583   3600#  3602   3619#
                    3621   3638#  3640   3657#  3659   3677#  3679   3694#  3696   3711#  3713   3728#  3730
                    3745#  3747   3762#  3764   3780#  3782   3799#  3801   3818#  3820   3837#  3839   3856#
                    3858   3875#  3877   3895#  3897   3912#  3914   3929#  3931   3946#  3948   3963#  3965
                    3980#  3982   3997#  3999   4014#  4016   4031#  4033   4048#  4050   4065#  4067   4083#
                    4085   4102#  4104   4121#  4123   4140#  4142   4159#  4161   4178#  4180   4197#  4199
                    4216#  4218   4235#  4237   4254#  4256   4273#  4275
$OCNT   025572      5270*  5299*  5312#
$OMODE  025574      5265*  5269*  5274   5277*  5288*  5314#
$OVER   024124      4926   4942   4950   4960   4969#
$PASS   001332       330#   547#   599   4347*  4348*  4956   4973
$PASTM  001006       229#
$POWER  026024      5389   5396#
$PWRAD  026012      5391#
$PWRON  025644       523   5356#  5386
$PWRMG  026006      5389#
$PWRUP  025716      5366   5372#
$QUES   001320       318#  5034   5181
$RDCHR= ****** U    5352
$RDDEC= ****** U    5352
$RDLIN= ****** U    5352
$RDOCT= ****** U    5352
$REGAD  001166       273#
$REGO   001170       275#   642*   643*   644   645*   647   664*   665*   666   667*   669   681*   682
                     697*   698    713*   714   729*   730   744*   746    752   766*   768   780    794*
                     796    803    813    827*  829    840   854*   856    862   867    881*  883    894
                     908*   910    917    931*  933    946   960*   962    978   992*   994   999    1013*
                    1015   1020   1034*  1036   1047   1070*  1071*  1072   1081  2009*  2014  2019   2041*
                    2046   2051   2056   2078*  2083   2096   2118*  2123   2133  2155*  2160  2166   2171
                    2193*  2198   2210   2232*  2237   2249   2271*  2276   2282  2304*  2309  2327   2343*
                    2348   2359   2381*  2386   2391   2413*  2418   2430   2452*  2455   2466  2483*  2486
                    2498   2515*  2518   2536   2553*  2556   2563   2580*  2583   2593   2610*  2613   2620
                    2637*  2640   2645   2650   2667*  2670   2677   2687   2704*  2707   2718   2735*  2738
                    2742   2759*  2762   2774   2789*  2792   2798   2820*  2822   2825   2828   2830   2832
                    2834   2835*  4402   4419*  4423*  4427*  4431*  4437*  4443*  4477   4496*  4502*  4508*
                    4514*  4523*  4532*  4545*  4549*  4567   4588*  4594*  4612   4636*  4640*  4658   4679*
                    4685*  4703   4727*  4731*  4749   4770*  4776*  4794   4819*  4823   4841   4852*  4868
                    4886   5893   5894   5895   5906   5910   5914   5919   5921   5923   5925   5927   5929
                    5931   5933   5935   5966   5968   5970   5973   5975   5979   5983   5987   5991
$REG1   001172       276#   646*   647   668*   669   749   752   771   780   799   813   833*   840
                     860*   867    887*  894    913   917   944*  946   970   973   978   996    999
                    1017   1020   1039   1042   1047  1079*  1080*  1081  2010*  2021  2042*  2058   2079*
                    2098   2119*  2135   2156*  2173  2194*  2212   2233*  2251  2272*  2284   2305*  2344*
                    2361   2382*  2393   2414*  2432  2453*  2468   2484*  2500  2516*  2538   2554*  2565
                    2581*  2595   2611*  2622   2638*  2652  2668*  2689   2705*  2720   2736*  2744   2760*
                    2790*  2800   2821*  2824*  2827*  2829*  2831*  2833*  2836*  2840   2843*  2847   2850*
```

```
                    2854   2857*  2861   2864*  2869   2872*  2877   4404   4479   4546*  4569   4589*  4614
                    4637*  4660   4680*  4705   4728*  4751   4771*  4796   4820*  4843   4863*  4888   5894
                    5895   5906   5910   5914   5919   5921   5923   5925   5927   5929   5931   5933   5935
                    5966   5968   5970   5973   5975   5979   5983   5987   5991
$REG10  001210       283#  2083*  2084   2198*  2199   2205   2245*  2246   2276*  2277   2309*  2310   2322
                    2421*  2422   2425   5949   5952   5955   5960   5964
$REG11  001212       284#  2087*  2088   2091   2202*  2203*  5960   5964
$REG12  001214       285#
$REG13  001216       286#
$REG14  001220       287#
$REG15  001222       288#
$REG16  001224       289#
$REG17  001226       290#
$REG2   001174       277#   749*   750*   771*   772    775    799*   800    808    829*   830    835    883*
                     884    889    913*   914    933*   934   1036*  1037   2011*  2023   2043*  2060   2080*
                    2100   2120*  2137   2157*  2175   2195*  2214   2234*  2253   2273*  2286   2306*  2345*
                    2363   2383*  2395   2415*  2434   2458   2461   2466   2490   2498   2522   2536   2560
                    2563   2585   2588   2593   2616   2620   2643*  2650   2673   2687   2715   2718   2740*
                    2742   2772*  2774   2796*  2798   2837*  2838   2844*  2845   2851*  2852   2858*  2859
                    2866*  2867   2874*  2875   4406   4481   4590*  4616   4681*  4707   4772*  4798   4864*
                    4890   5895   5910   5914   5925   5927   5929   5931   5933   5935   5966   5968   5970
                    5973   5975   5979   5983   5987   5991
$REG3   001176       278#  2012*  2025   2044*  2062   2081*  2102   2121*  2139   2158*  2177   2196*  2216
                    2235*  2255   2274*  2288   2307*  2346*  2365   2384*  2397   2416*  2436   2468   2500
                    2538   2565   2595   2622   2652   2689   2720   2744   2800   2840   2847   2854   2861
                    2869   2877   4408   4483   4531*  4618   4682*  4709   4773*  4800   4865*  4892   5895
                    5910   5914   5966   5968   5970   5973   5975   5979   5983   5987   5991
$REG4   001200       279#  2017*  2019   2049*  2056   2087   2096   2125   2128   2133   2163   2171   2202
                    2210   2245   2249   2279   2282   2320*  2327   2351   2354   2359   2389*  2391   2421
                    2430   2486*  2487   2493   2518*  2519   2526   2556*  2557   2613*  2614   2673*  2674
                    2682   2715*  2716*  5937   5939   5941   5943   5946   5949   5952   5955   5958   5962
                    5973   5977   5981   5985   5989   5993
$REG5   001202       280#  2021   2058   2098   2135   2173   2212   2251   2284   2361   2393   2432   2522*
                    2523   2531   2616*  2617   5937   5939   5941   5943   5946   5949   5952   5955   5958
                    5962   5989   5993
$REG6   001204       281#  2023   2060   2100   2137   2175   2214   2253   2286   2363   2395   2434   5937
                    5939   5941   5943   5946   5949   5952   5955   5958   5962
$REG7   001206       282#  2025   2062   2102   2139   2177   2216   2255   2288   2365   2397   2436   5937
                    5939   5941   5943   5946   5949   5952   5955   5958   5962
$R2A    = ****** U   5352
$SAVRE  = ****** U   5352
$SAVR6  026022       5365*  5373   5374*  5375*  5395#
$SCOPE  023664        517   4923#
$SETUP= 000037        502#   516    517    519    521    523    525    526    527    529    4924   4988   5022
                     5029
$STUP   = 177777      502#
$SVLAD  024070       4934   4963#
$SVPC   = 000204      203#   208
$SWR    = 167400       1#    12     315    316    317    526    527    529    530    639    661    679    695
                      711    727    744    766    794    827    854    881    908    931    960    992   1013
                     1034   1066   1096   1114   1132   1150   1168   1186   1204   1216   1228   1240   1252
                     1264   1277   1289   1301   1313   1325   1337   1350   1362   1374   1386   1398   1410
                     1423   1435   1447   1459   1471   1484   1496   1508   1520   1532   1546   1565   1584
                     1603   1622   1641   1661   1673   1685   1697   1709   1721   1734   1746   1758   1770
                     1782   1794   1807   1819   1831   1843   1855   1867   1880   1892   1904   1916   1928
                     1941   1953   1965   1977   1989   2008   2040   2077   2117   2154   2192   2231   2270
```

```
                    2303    2342    2380    2412    2451    2482    2514    2552    2579    2609    2636    2666    2703
                    2734    2758    2788    2819    2886    2903    2920    2937    2954    2971    2988    3005    3022
                    3039    3056    3074    3093    3112    3131    3150    3169    3188    3207    3226    3245    3264
                    3284    3301    3318    3335    3352    3369    3386    3403    3420    3437    3454    3472    3491
                    3510    3529    3548    3567    3586    3605    3624    3643    3662    3682    3699    3716    3733
                    3750    3767    3785    3804    3823    3842    3861    3880    3900    3917    3934    3951    3968
                    3985    4002    4019    4036    4053    4070    4088    4107    4176    4145    4164    4183    4202
                    4221    4240    4259    4278    4915    4916    4917    4918    4919    4925    4937    4939    4940
                    4943    4944    4945    4952    4953    4954    4966    4969    4972    4979    4980    4981    4982
                    4983    5000    5007    5019    5022    5034    5392
$SWREG  001346       338#    550
$SWRMK= 000000      4919    4941
$TESTN  001330       329#   4964*   5095
$TIMES  001310       315#    526*   4298*   4346*   4952*   4959    4962*   4972
$TKB    001154       266#
$TKS    001152       265#
$TMP0   001230       291#   4376    4451    4539    4582    4630    4673    4721    4764    4813    4856
$TMP1   001232       292#
$TMP10  001250       299#   5912
$TMP11  001252       300#   5898    5912
$TMP12  001254       301#   5902
$TMP13  001256       302#   5899
$TMP14  001260       303#   5904
$TMP15  001262       304#
$TMP16  001264       305#
$TMP17  001266       306#
$TMP2   001234       293#   5906
$TMP20  001270       307#
$TMP21  001272       308#
$TMP22  001274       309#
$TMP23  001276       310#
$TMP24  001300       311#
$TMP25  001302       312#
$TMP26  001304       313#
$TMP27  001306       314#
$TMP3   001236       294#   5906
$TMP4   001240       295#   5908
$TMP5   001242       296#   5897    5908
$TMP6   001244       297#   5900    5908    5912
$TMP7   001246       298#   5908    5912
$TN     = 000277       1#     12     635    639#    657    661#    675    679#    691    695#    707    711#    723
                     727#    740    744#    757    762    766#    785    790    794#    818    823    827#    845
                     850     854#    872    877     881#    899    904    908#    922    927    931#    951    956
                     960#    983    988    992#   1004   1009   1013#   1025   1030   1034#   1052   1062   1066#
                    1086   1092   1096#   1100   1110   1114#   1118   1128   1132#   1136   1146   1150#   1154
                    1164   1168#   1172   1182   1186#   1190   1200   1204#   1208   1212   1216#   1220   1224
                    1228#   1232   1236   1240#   1244   1248   1252#   1256   1260   1264#   1268   1273   1277#
                    1281   1285   1289#   1293   1297   1301#   1305   1309   1313#   1317   1321   1325#   1329
                    1333   1337#   1341   1346   1350#   1354   1358   1362#   1366   1370   1374#   1378   1382
                    1386#   1390   1394   1398#   1402   1406   1410#   1414   1419   1423#   1427   1431   1435#
                    1439   1443   1447#   1451   1455   1459#   1463   1467   1471#   1475   1480   1484#   1488
                    1492   1496#   1500   1504   1508#   1512   1516   1520#   1524   1528   1532#   1536   1542
                    1546#   1550   1561   1565#   1569   1580   1584#   1588   1599   1603#   1607   1618   1622#
                    1626   1637   1641#   1645   1657   1661#   1665   1669   1673#   1677   1681   1685#   1689
                    1693   1697#   1701   1705   1709#   1713   1717   1721#   1725   1730   1734#   1738   1742
                    1746#   1750   1754   1758#   1762   1766   1770#   1774   1778   1782#   1786   1790   1794#
```

```
                    1798    1803    1807#   1811    1815    1819#   1823    1827    1831#   1835    1839    1843#   1847
                    1851    1855#   1859    1863    1867#   1871    1876    1880#   1884    1888    1892#   1896    1900
                    1904#   1908    1912    1916#   1920    1924    1928#   1932    1937    1941#   1945    1949    1953#
                    1957    1961    1965#   1969    1973    1977#   1981    1985    1989#   1993    2004    2008#   2030
                    2036    2040#   2067    2073    2077#   2107    2113    2117#   2144    2150    2154#   2182    2188
                    2192#   2221    2227    2231#   2260    2266    2270#   2293    2299    2303#   2332    2338    2342#
                    2370    2376    2380#   2402    2408    2412#   2441    2447    2451#   2473    2478    2482#   2506
                    2510    2514#   2543    2548    2552#   2570    2575    2579#   2600    2605    2609#   2627    2632
                    2636#   2657    2662    2666#   2694    2699    2703#   2725    2730    2734#   2749    2754    2758#
                    2779    2784    2788#   2805    2815    2819#   2881    2886#   2889    2898    2903#   2906    2915
                    2920#   2923    2932    2937#   2940    2949    2954#   2957    2966    2971#   2974    2983    2988#
                    2991    3000    3005#   3008    3017    3022#   3025    3034    3039#   3042    3051    3056#   3059
                    3069    3074#   3077    3088    3093#   3096    3107    3112#   3115    3126    3131#   3134    3145
                    3150#   3153    3164    3169#   3172    3183    3188#   3191    3202    3207#   3210    3221    3226#
                    3229    3240    3245#   3248    3259    3264#   3267    3279    3284#   3287    3296    3301#   3304
                    3313    3318#   3321    3330    3335#   3338    3347    3352#   3355    3364    3369#   3372    3381
                    3386#   3389    3398    3403#   3406    3415    3420#   3423    3432    3437#   3440    3449    3454#
                    3457    3467    3472#   3475    3486    3491#   3494    3505    3510#   3513    3524    3529#   3532
                    3543    3548#   3551    3562    3567#   3570    3581    3586#   3589    3600    3605#   3608    3619
                    3624#   3627    3638    3643#   3646    3657    3662#   3665    3677    3682#   3685    3694    3699#
                    3702    3711    3716#   3719    3728    3733#   3736    3745    3750#   3753    3762    3767#   3770
                    3780    3785#   3788    3799    3804#   3807    3818    3823#   3826    3837    3842#   3845    3856
                    3861#   3864    3875    3880#   3883    3895    3900#   3903    3912    3917#   3920    3929    3934#
                    3937    3946    3951#   3954    3963    3968#   3971    3980    3985#   3988    3997    4002#   4005
                    4014    4019#   4022    4031    4036#   4039    4048    4053#   4056    4065    4070#   4073    4083
                    4088#   4091    4102    4107#   4110    4121    4126#   4129    4140    4145#   4148    4159    4164#
                    4167    4178    4183#   4186    4197    4202#   4205    4216    4221#   4224    4235    4240#   4243
                    4254    4259#   4262    4273    4278#   4281    4293
$TPB    001160       268#    5170*   5181
$TPFLG  001165       272#    5119    5181
$TPS    001156       267#    5168    5181
$TRAP   025576       521     5323#
$TRAP2  025620      5334#    5345
$TRP  = 000005      5338#    5347#   5348#   5349#   5350#
$TRPAD  025632      5328     5345#
$TSTM   001004       228#
$TSTNM  001102       243#    4300*   4345*   4941    4963*   4964    4969    4973    4999    5034
$TYPBN= ****** U    5350
$TYPDS= ****** U    5350
$TYPE   024620      5119#    5212    5338    5346
$TYPEC  025032      5149     5156    5163    5168#   5169
$TYPER  024404      5009     5048#
$TYPEX  025100      5174     5176    5179#
$TYPOC  025374      5268#    5347
$TYPON  025410      5267     5270#   5349
$TYPOS  025350      5263#    5348
$UNIT   001336       332#
$UNITM  001010       230#
$USWR   001350       339#
$XTSTR  023674      4928#
$OFILL  025573      5264*    5268*   5278    5313#
$4OCAT= ******* U   4925     5009
.     = 034074       189#     193#    203     204#    206#    208#    209#    215     216#    218#    220#    239#    321
                     503#     514     529     530    4378    4453    4541    4584    4632    4675    4723    4766    4815
                    4858     4972    4973    5034    5101#   5181    5233#   5368    5394
.$ASTA= ****** U    5185     5188
```

```
.$X    = 001000            215#    220
```

```
CMPFLT      1#  4402  4477  4567  4612  4658  4703  4749  4794  4841  4886
COMMEN    141#
COMM00      1#
COMM01      1#
COMM02      1#
COMM03      1#
COMM04      1#
COMM05      1#
COMM06      1#
COMM07      1#
COMM1       1#  2917  2934  2985  3002  3071  3128  3147  3204  3223  3281  3315  3349  3383  3488
         3526   3564  3602  3621  3640  3696  3730  3764  3782  3839  3858  3897  3931  3965  3999
         4033   4104  4142  4180  4218  4256
COMM10      1#
COMM11      1#
COMM12      1#
COMM13      1#
COMM14      1#
COMM15      1#
COMM16      1#
COMM17      1#
COMM2       1#  3053  3451  4067
COMM20      1#
COMM21      1#
COMM22      1#
COMM23      1#
COMM24      1#
COMM25      1#
COMM26      1#
COMM27      1#
COMM3       1#  2883  2900  2951  2968  3019  3036  3090  3109  3166  3185  3242  3298  3332  3366
         3400   3417  3434  3469  3507  3545  3583  3679  3713  3747  3801  3820  3877  3914  3948
         3982   4016  4050  4085  4123  4161  4199  4237
COMM30      1#
COMM31      1#
COMM32      1#
COMM33      1#
COMM34      1#
COMM35      1#
COMM36      1#
COMM37      1#
COMM4       1#  3261  3659  4275
COMM40      1#
COMM41      1#
COMM42      1#
COMM43      1#
COMM44      1#
COMM45      1#
COMM46      1#
COMM47      1#
ENDCOM    141#
ERRCMP      1#   752   775   780   803   808   813   835   840   862   867   889   894   917   946
          973    978   999  1020  1042  1047  2051  2091  2128  2166  2205  2322  2327  2354  2425
         2461   2493  2526  2531  2588  2645  2677  2682  2774
ERRLUR      1#  5033
ERROR      35#   185   649   671   684   700   716   732   754   777   782   805   810   815   837
```