FP11-F FP11F FLTG PNT PRT A AH-F632B-MC
CKFPAB0 FICHE 1 OF 1
FEB 1981

MADE IN USA

```
                .REM    &

                        IDENTIFICATION

                PRODUCT CODE:   AC-F630B-MC
                PRODUCT NAME:   CKFPAB0 FP11F FLTG PNT PRT A
                DATE CREATED:   OCTOBER, 1980
                MAINTAINER:     DIAGNOSTIC ENGINEERING
                AUTHOR:         DAN MILLEVILLE
```

HISTORY

NO CHANGES TO THE 11/34 FLOATING POINT DIAGNOSTIC PART 'A' WERE FOUND TO BE NEEDED TO ADAPT IT FOR USE ON THE 11/44.

THE FOLLOWING WAS ADDED TO THE 11/34 FLOATING POINT DIAGNOSTIC TO MAKE THE 'B' VERSION COVER THE 11/44:

1. TEST 22 - PROCESSOR LOOKS TO SEE IF APT IS CONTROLLING THE TEST, AND IF IT IS, CHECKS TO SEE IF THE USER HAS SELECTED THIS TEST BY CHECKING BIT 7 IN THE SWITCH REGISTER. IT HAS ALSO BEEN CHANGED SO THAT IF BIT 7 IS *ONE*, THE CODE WILL SELECT THE TEST.

THE FOLLOWING WAS ADDED TO THE 11/34 FLOATING POINT DIAGNOSTIC TO MAKE THE 'C' VERSION COVER THE 11/44:

1. TEST 76 - CHECKS THAT FP PROCESSOR DOESN'T ACCESS D-SPACE UNTIL CONDITIONS WARRANT.

2. TEST 77 - CHECKS THAT SR1 MATCHES WHAT ACTUALLY HAPPENED TO THE REGISTER OF THE INSTRUCTION, AND THAT THE VALUE OF AUTO INCREMENT/DECREMENT WAS PROPER.

CONTENTS

1. ABSTRACT

THE THREE PROGRAMS:

CKFPAB0 CKFPBA0 CKFPCA0

ARE DESIGN TO DETECT AND REPORT LOGIC FAULTS IN THE PDP 11/44 FP11-F FLOATING POINT PROCESSOR. THE DESIGN IS AN ATTEMPT TO REACH ALL ROM STATES, TAKE ALL BRANCH MICRO TESTS (BUT'S) AND VERIFY ALL THE LOGIC. THEY CONSIST OF 157 (OCT) INDIVIDUAL TESTS SEQUENCED TO DETECT AND ATTEMPT TO IDENTIFY FAULTS WITH A MINIMUM HARDWARE OR SOFTWARE LEVEL. THE TESTS ARE PARTIONED INTO THREE STAND-ALONE PROGRAMS DESCRIBED BELOW.

NOTE THAT ERROR REPORTS IN THESE PROGRAMS ARE BASED UPON THE KNOWLEDGE THAT ALL PREVIOUS TESTS HAVE BEEN RUN AND IN MOST CASE THAT THERE IS ONLY A SINGLE POINT FAULT IN THE FP11-F. IF THE PROGRAMS OR TESTS ARE NOT RUN IN ORDER THEN ERROR MESSAGES MAY NOT BE ACCURATE.

A. CKFPAB0

CKFPAB0 TESTS:

- LDFPS
- STFPS
- CFCC
- SETF, SETD, SETI AND SETL
- STST
- LDF AND LDD (ALL SOURCE MODES)
- STD (MODE 0 AND 1)
- ADDF, ADDD AND SUBD (MOST CONDITIONS)

B. CKFPBA0

CKFPBA0 TESTS:

- ADDF, ADDD AND SUBD (ALL CONDITIONS NOT TESTED IN CKFPAB0)
- CMPD AND CMPF
- DIVD AND DIVF
- MULD AND MULF
- MODD AND MODF

C. CKFPCA0

CKFPCA0 TESTS:

STF AND STD (ALL MODES)
STCFD AND STCDF
CLRD AND CLRF
NEGF AND NEGD
ABSF AND ABSD
TSTF AND TSTD
NEGF, ABSF AND TSTF (ALL SOURCE MODES)
NEGF, ABSF AND TSTF (ALL SOURCE MODES)
LDFPS (ALL SOURCE MODES)
LDCIF AND LDCLF
LDCID AND LDCLD
LDEXP
STFPS (ALL DESTINATION MODES)
STCFL AND STCFI
STCDL AND STCDI
STEXP
STST

2. REQUIREMENTS

2.1 EQUIPMENT

A PDP 11/44 (WITH OR WITHOUT CONSOLE), LA30 (OR EQUIVALENT) AND AN FP11-F FLOATING POINT PROCESSOR. NOTE THAT A SPECIAL INTERRUPTS TEST MODULE IS BEING DESIGNED FOR USE IN THE MANUFACTURING ENVIRONMENT. WHEN THIS DEVICE IS PRESENT THE PROGRAM CKFPBA0 WILL MAKE USE OF IT TO TEST THE FPP INTERRUPT ON BUS REQUEST FUNCTIONS.

2.2 STORAGE

ALL THREE PROGRAM REQUIRE A MEMORY SYSTEM OF AT LEAST 16K TO LOAD AND RUN.

2.3 PRELIMINARY PROGRAMS

THESE THREE DIAGNOSTICS WILL ASSUME THAT THE PDP 11/44 CENTRAL PROCESSOR IS FAULTLESS, THEREFORE WHEN IN DOUBT RUN THE PDP 11/44 PROCESSOR DIAGNOSTICS BEFORE THESE FP11-F DIAGNOSTICS.

3. LOADING PROCEDURE

THE PROGRAMS WILL BE SUPPLIED ON THE 11/44 DIAGNOSTIC MEDIA. REFER TO THE XXDP OPERATING MANUAL FOR FURTHER INFORMATION.

4. STARTING PROCEDURE

4.1 CONTROL SWITCH SETTINGS

SEE SECTION 5.1

4.2 PROGRAM AND OPERATOR ACTION

1. LOAD PROGRAM INTO MEMORY
2. LOAD ADDRESS 200
3. SET CONSOLE SWITCHES (IF CONSOLE IS PRESENT)
4. PRESS START
   ON FIRST PASS THE PROGRAM WILL IDENTIFY ITSELF. NOTE THAT IF THERE IS NO PHYSICAL CONSOLE THE PROGRAM WILL REQUEST THE OPERATOR FOR INITIAL VALUE FOR THE SOFTWARE SWITCH REGISTER (SEE SECTION 8.5). IF RUNNING UNDER ACT, APT OR CHAIN THIS DOES NOT APPLY.
5. THE PROGRAM WILL LOOP AND AN END OF PASS AND ERROR SUMMARY WILL BE TYPED AT THE END OF EVERY PASS.

5. OPERATING PROCEDURE

5.1 OPERATIONAL SWITCH SETTINGS

THE SWITCH SETTING ARE:

	OCIAL	
SW<15>=1...	100000	HALT ON ERROR
SW<14>=1...	40000	LOOP ON CURRENT TEST
SW<13>=1...	20000	INHIBIT ERROR TYPE OUTS
SW<12>=1...	10000	INHIBIT T-BIT TRAPPING
SW<11>=1...	4000	INHIBIT ITERATIONS
SW<10>=1...	2000	RING TTY BELL ON ERROR
SW<9>=1....	1000	LOOP ON ERROR
SW<8>=1....	400	LOOP ON TEST SPECIFIED IN SW<6> THROUGH SW<0>
SW<7>=1....	200	PRINT ERROR SUMMARY EVEN IF SW<13>=1, THIS APPLIES ONLY TO PROGRAM CKFPAB0.
SW<7>=1....	200	SELECT CORRECT INTERRUPT TEST IN PROGRAM CKFPBA0.

6. ERRORS

6.1 SUMMARIES

IN PROGRAM CKFPA0 TESTS 1 AND 11 HAVE A SPECIAL ERROR SUMMARY FEATURE. THESE TWO TEST RUN MANY TEST PATTERNS THROUGH THE LOGIC. AFTER AN ERROR IS ENCOUNTERED, ONLY THE FIRST FIVE ERRORS ARE REPORTED (TYPED ON THE TTY). EVERY ERROR THOUGH IS LOGGED AND AN ERROR SUMMARY IS PRINTED WHEN THE TEST IS COMPLETE. NOTE THAT IF SW<13>=1, THIS SUMMARY WILL NOT BE TYPED UNLESS SW<7>=1. IN OTHER WORDS TO GET JUST AN ERROR SUMMARY FROM EITHER OF THESE TWO TESTS 1 AND 11 IN PROGRAM CKFPAB0 BOTH SWITCHES 13 AND 7 MUST = 1.

6.2 ERROR RECOVERY

SW<15:9>=0... MOST ERRORS WILL CAUSE EXECUTION TO GO TO THE START OF THE NEXT TEST AFTER THE MESSAGE IS TYPED. A FEW TESTS ARE IN SECTIONS. IN THESE TESTS AN ERROR WILL CAUSE EXECUTION TO GO TO THE NEXT SECTION AFTER THE MESSAGE IS TYPED.

SW<15>=1... THE PROGRAM WILL HALT AFTER TYPING THE ERROR MESSAGE. PRESSING THE CONSOLE CONTINUE WILL CAUSE THE PROGRAM TO CONTINUE AS IF SW<15>=0.

7. RESTRICTIONS

NONE

8. MISCELLANEOUS

8.1 EXECUTION TIMES

LESS THAN 10 SECONDS FOR EACH PROGRAM ON ANY PASS.

8.2 STACK POINTER

THE STACK POINTER IS INITIALIZED TO 1100 IN EACH OF THE THREE PROGRAMS.

8.3 PASS COUNT

THE PROGRAM MAKES ONE PASS FOR EACH END OF PASS MESSAGE TYPED. THE END OF PASS MESSAGE DESCRIBES THE TOTAL NUMBER OF PASSES COMPLETED AND THE TOTAL NUMBER OF ERRORS SINCE THE LAST END OF PASS MESSAGE.

8.4 T-BIT TRAPPING

IF SW<12>=0 EACH PROGRAM WILL RUN WITH TRACE TRAPS ON EVERY OTHER PASS. FIRST PASS WILL NOT ENABLE TRACE TRAPS. NOTE SW<12>=1 DISABLES T-BIT TRAPS.

8.5 SOFTWARE SWITCH REGISTER

EACH OF THE THREE PROGRAMS WILL RUN WITH OR WITHOUT A CONSOLE SWITCH REGISTER. IF A PHYSICAL CONSOLE SWITCH REGISTER IS PRESENT ON THE SYSTEM, THEN THESE PROGRAMS WILL GO AHEAD AND USE IT FOR THE SWITCH FUNCTIONS DESCRIBED IN 5.1 ABOVE. IF HOWEVER THERE IS NO CONSOLE SWITCH REGISTER ON THE SYSTEM A SOFTWARE SWITCH REGISTER WILL BE USED. THIS SOFTWARE SWITCH REGISTER CAN BE EXAMINED OR MODIFIED AT ANY TIME BY THE USER IF HE TYPES CONTROL G WHILE THE PROGRAM IS RUNNING. THIS CONTROL G WILL CAUSE THE CONTENTS OF THE SOFTWARE SWITCH REGISTER TO BE TYPED ON THE TTY AND ASK THE USER FOR A NEW VALUE. WHEN THE USER TYPES A VALUE AND CARRIAGE RETURN THEN THE PROGRAM WILL RESUME TESTING AT THE SAME POINT AT WHICH IT LEFT OFF WHEN THE USER TYPED CONTROL G. NOTE THAT WHEN NOT RUNNING UNDER ACT, APT OR CHAIN THE USER WILL BE ASKED FOR A SOFTWARE SWITCH REGISTER VALUE AFTER LOADING ADDRESS 200 AND STARTING THE PROGRAM THE FIRST TIME THE PROGRAM IS RUN AFTER LOADING (ONLY IF NO CONSOLE SWITCH REGISTER IS ON THE SYSTEM).

8.6 INTERRUPTS TEST

IN PROGRAM CKFPBA0 THERE IS A SPECIAL TEST FOR CHECKING THE CORRECT FLOWS OF THE FPP. THIS TEST CAN BE RUN ONLY IF A SPECIAL TEST MODULE IS IN THE SYSTEM. THIS MODULE WILL PROBABLY ONLY BE USED IN MANUFACTURING. IF THIS MODULE IS NOT IN THE SYSTEM THIS TEST WILL AUTOMATICALLY BE DESELECTED. IF THIS TEST MODULE IS ON THE SYSTEM AND SW<7>=1 THIS TEST WILL BE RUN. IF SW<7>=0 THIS TEST WILL BE DESELECTED.

8.7 ACT, APT AND XXDP COMPATIBILTY

THESE PROGRAMS ARE FULLY COMPATIBLE WITH:

- APT
- ACT
- XXDP MONITOR AND CHAIN PROGRAMS.

9. PROGRAM DESCRIPTION

TEST 1 LDFPS, STFPS AND DATA PATHS TEST

THIS IS A TEST OF THE LDFPS (LOAD FLOATING POINT STATUS) AND STFPS (STORE FLOATING POINT STATUS) INSTRUCTIONS. A COUNT PATTERN IS GENERATED AND RUN THROUGH THE FLOATING POINT STATUS REGISTER. THIS WILL TEST THE 16-BIT TRI STATE BUS WHICH CONNECTS THE CPU WITH THE FPP AND ALSO RUNS INTERNALLY WITHIN THE FPP. ONLY DMO AND SMO ARE USED. NOTE THAT A MASK MUST BE USED BECAUSE SOME OF THE FPS BITS CANNOT BE SET.

ONLY THE FIRST FIVE ERRORS WILL BE REPORTED INDIVIDUALLY. THIS IS TO PREVENT LOCKING OUT THE COMPLETION OF THE TEST BECAUSE OF VIRTUALLY ENDLESS NUMBER OF ERRORS. ONLY FIVE INDIVIDUAL ERRORS WILL BE REPORTED THEN THE TEST WILL BE COMPLETED AND AN ERROR SUMMARY GIVEN (SEE NOTE BELOW).

NOTE THAT THIS TEST KEEPS A DYNAMIC RECORD OF THE LOGICAL 'AND' AND 'OR' OF THE FAILING DATA PATTERNS. THESE CAN BE VERY USEFUL IN DETERMINING STUCK BITS. IF THE USER HAS THE INHIBIT ERROR TYPE OUT SWITCH (SWR13) OFF, THEN THE USER WILL RECEIVE EACH INDIVIDUAL ERROR MESSAGE PLUS AN ERROR SUMMARY AT THE END OF THE TEST. INHIBITING ERROR PRINT OUT WILL INHIBIT ERROR SUMMARY PRINT OUT, EXCEPT IN THE CASE DESCRIBED BELOW. TO GET JUST THE ERROR SUMMARY WITH NO INDIVIDUAL ERROR REPORTS, SET SWITCH REGISTER BIT13 AND SWITCH REGISTER BIT7 BOTH ON.

TEST 2 CFCC TEST

THIS IS A TEST OF THE COPY CONDITION CODES INSTRUCTION, CFCC.

TEST 3 SETF, SETD, SETI AND SETL TEST

THIS IS A TEST OF THE SETF, SETD, SETI AND SETL INSTRUCTIONS. EACH INSTRUCTION IS EXECUTED WITH THE FPS CONTAINING ALL ONES AND ALSO WITH THE FPS CLEAR. THE RESULT OF EACH SITUATION IS CHECKED.

```
                                TEST 4          ILLEGAL FPP OP CODES AND STST TEST
                                ------          ----------------------------------

                                        THIS IS A TEST OF THE FPP OPERATION CODES:

                                                                170003
                                                                170004
                                                                  :
                                                                170010
                                                                170013
                                                                170014
                                                                  :
                                                                170077

                                        THESE  ARE ILLEGAL INSTRUCTIONS AND WITH  INTERRUPTS
                                        ENABLED  SHOULD  CAUSE  A  TRAP TO 244.  ALSO TESTED
                                        HERE IS THE INSTRUCTION:  STST R1, WHICH SHOULD  PUT
                                        THE  FEC  CODE  2  IN  R1, AFTER ANY OF THE ABOVE OP
                                        CODES IS EXECUTED.

                                TEST 5          FID, INTERRUPT DISABLE, BIT TEST
                                ------          --------------------------------

                                        THIS IS A TEST OF  FPS  BIT  14  (FID)  OR  FLOATING
                                        INTERRUPT   DISABLE.    AN  ILLEGAL  INSTRUCTION  IS
                                        EXECUTED WITH FID=1.  NO INTERRUPT SHOULD OCCUR.

                                TEST 6          LDD AND STD, WITH SRC AND DST MODE 1, TEST
                                ------          ------------------------------------------

                                        THIS IS A TEST OF BOTH THE INSTRUCTION:
                                                        LDD     (R0),AC0
                                        AND THE INSTRUCTION:
                                                        STD     AC0,(R0)          MOST OF THE
                                        FAILURES ARE ISOLATED TO THE SRC OR DST FLOWS.  NOTE
                                        THAT THE INTEGRITY OF  AC0  HAS  NOT  BEEN  ASSURED.
                                        THIS  MEANS THAT IN SOME CASES IT WILL BE IMPOSSIBLE
                                        TO ISOLATE CERTAIN DATA PATTERN FAILURES  TO  EITHER
                                        THE FLOWS OR THIS ACCUMULATOR.

                                TEST 7          FSRC MODE 0 TEST
                                ------          ----------------

                                        THIS IS A TEST OF FSRC MODE ZERO USING THE  LDD  AND
                                        LDF INSTRUCTIONS.


                                TEST 10         FDST MODE 0 TEST
                                -------         ----------------

                                        THIS IS A TEST OF THE STORE  INSTRUCTIONS,  STD  AND
                                        STF, WITH FDST MODE 0.
```

```
                                TEST 11         ACCUMULATORS DATA PATTERNS TEST
                                -------         -------------------------------

                        THIS IS A  TEST  OF  THE  FLOATING  POINT  PROCESSOR
                        ACCUMULATORS.

                        EACH ACCUMULATOR IS TESTED IN TWO WAYS:
                                1       TEST PATTERN GENERATED BY FLOATING A
                                ONE ACROSS A FIELD OF ZEROES.
                                2       TEST PATTERN GENERATED BY FLOATING A
                                ZERO ACROSS A FIELD OF ONES.
                        EACH OF ACCUMULATORS AC0 THROUGH AC5 IS TESTED.

                        NOTE THAT THIS TEST KEEPS A DYNAMIC  RECORD  OF  THE
                        LOGICAL 'AND' AND 'OR' OF THE FAILING DATA PATTERNS.
                        THESE CAN BE VERY USEFUL IN DETERMINING STUCK  BITS.
                        IF  THE  USER  HAS THE INHIBIT ERROR TYPE OUT SWITCH
                        (SWR13)  OFF,  THEN  THE  USER  WILL  RECEIVE   EACH
                        INDIVIDUAL  ERROR  MESSAGE  PLUS AN ERROR SUMMARY AT
                        THE END OF THE TEST.   INHIBITING  ERROR  PRINT  OUT
                        WILL  INHIBIT ERROR SUMMARY PRINT OUT, EXCEPT IN THE
                        CASE DESCRIBED BELOW.  TO GET JUST THE ERROR SUMMARY
                        WITH   NO   INDIVIDUAL  ERROR  REPORTS,  SET  SWITCH
                        REGISTER BIT13 AND SWITCH REGISTER BIT7 BOTH ON.


                        THE FOLLOWING PROCEDURE  IS  PRESENTED  TO  AID  THE
                        TROUBLE  SHOOTER  IN  SITUATIONS  WHERE  AM2901 CHIP
                        ISOLATION IS ATTEMPTED.

                        WARNING:  THIS PROCEDURE ASSUMES THAT THE  FAULT  IS
                        IN  ONE OF THE AM2901 CHIPS.  THIS ASSUMPTION IS NOT
                        NECESSARILY VALID IN ALL SITUATIONS.  IT REMAINS  TO
                        BE    SEEN    WHAT    NUMBER    OF    FAILURES    CAN
                        PROBABLILISTICALLY ASSOCIATED WITH THEM.  NOTE  ALSO
                        THAT   THIS  INFORMATION  SHOULD  NOT  BE  TAKEN  AS
                        ABSOLUTE, THAT IS THIS INFORMATION IS  THE  AUTHOR'S
                        SUGGESTION  FOR  ACHIEVING ISOLATION WHEN CHIP LEVEL
                        REPAIR IS NECESSARY.

                        WHEN THIS TEST HAS FINISHED RUNNING, IF ERRORS  HAVE
                        OCCURRED,  AN  ERROR  SUMMARY  WILL  BE TYPED.  THUS
                        SUMMARY WILL CONSIST OF TWO IMPORTANT QUANTITIES:
                                A.      FOUR SIXTEEN BIT NUMBERS LABELED THE
                                        LOGICAL  'AND'  ('*') OF THE FAILING
                                        DATA PATTERNS.
                                B.      FOUR SIXTEEN BIT NUMBERS LABELED THE
                                        LOGICAL  'OR'  ('+')  OF THE FAILING
                                        DATA PATTERNS.
```

```
                        A BIT STUCK HIGH IN THE HARDWARE WILL SHOW UP AS A 0
                        IN THAT BIT POSITION OF THE 'OR' OF THE FAILING DATA
                        PATTERNS.

                        A BIT STUCK LOW IN THE HARDWARE WILL SHOW UP AS A 1
                        IN  THAT  BIT  POSITION  OF THE 'AND' OF THE FAILING
                        DATA PATTERNS.

                        THUS IF A FAILURE OCCURS:
                                A.      STUCK HIGHS WILL SHOW AS 0'S IN  THE
                                'OR' PATTERN.
                                B.      STUCK LOWS WILL SHOW AS 1'S  IN  THE
                                'AND' PATTERN.

                        IF THE FAILURE IS  INTERMITTANT  THEN  THIS  PROCEDURE  WILL
                        STILL  APPLY!!  IF THE FAILURE MOVES FROM ONE BIT TO
                        ANOTHER  OR FROM ONE GROUP OF BITS TO ANOTHER  GROUP
                        OF  BITS THEN THE FAULT WILL PROBABLY NOT SHOW UP IN
                        THE 'AND' OR THE 'OR' PATTERNS;  IN  THIS  CASE  THE
                        'AND'  PATTERN  WILL BE ALL 0'S AND THE 'OR' PATTERN
                        WILL BE ALL 1'S.  WHEN THIS OCCURS SOME OTHER METHOD
                        OF  REPAIR MUST BE FOUND (SUCH AS INSPECTION OF EACH
                        INDIVIDUAL  ERROR  REPORT  RATHER  THAN  USING   THE
                        SUMMARY).

                        MAP THE FOLLOWING NOTATION ONTO EACH BIT POSITION IN
                        THE  'AND'  AND THE 'OR' PATTERNS WHICH ARE TYPED IN
                        THE ERROR SUMMARY.

                        A15,A14,...A1,A0        B15,B14,...B1,B0
                                C15,C14,...C1,C0        D15,D14,...C1,C0

                        IN THIS NOTATION A15 THROUGH A0 IS THE FIRST OF  THE
                        FOUR  16  BIT OCTAL NUMBERS TYPED, B15 THROUGH B0 IS
                        THE SECOND, ETC.

                        THIS TABLE SHOWS THE CORRESPONDING AM2901 CHIP  ('E'
                        NUMBER)  WHICH  IS RESPONSIBLE FOR EACH BIT POSITION
                        USING THE ABOVE NOTATION.  NOTE THAT  ECO'S  TO  THE
                        HARDWARE MIGHT MAKE THIS TABLE OBSOLETE IF IT IS NOT
                        UP DATED.  NOTE ALSO THAT THERE ARE  FOUR  BITS  FOR
                        EACH AM2901 CHIP:
```

BITS	AM2901 CHIP NUMBER
A15,A14,A13,A12	E37
A11,A10,A9,A8	E45
A7,A6,A5,A4	E34
A3,A2,A1,A0	E42
B15,B14,B13,B12	E33
B11,B10,B9,B8	E41
B7,B6,B5,B4	E36
B3,B2,B1,B0	E44
C15,C14,C13,C12	E35
C11,C10,C9,C8	E43
C7,C6,C5,C4	E38
C3,C2,C1,C0	E46
D15,D14,D13,D12	E39
D11,D10,D9,D8	E47
D7,D6,D5,D4	E40
D3,D2,D1,D0	E48

NOW FIVE IMPORTANT CASES WHICH WILL ARRISE WHEN A FAULTY AM2901 IS PRESENT CAN BE DESCRIBED:

1.) IF ONLY ONE BIT OF THE 64 BITS IS INCORRECT THE CHIP INDICATED IN THE ABOVE TABLE IS MOST PROBABLY AT FAULT. BUT IF THAT CHIP IS REPLACED AND THE ERROR PERSISTS THEN SUPPOSE THAT BIT IS,

LN WHERE 'L' IS A, B, C OR D
AND N IS 15, 14, ... OR 0

THEN IN GENERAL ANY OF THE FOUR CHIPS RESPONSIBLE FOR AN, BN, CN OR DN COULD BE AT FAULT, WITH LN BEING MOST PROBABLE.

FOR EXAMPLE IF BIT C12 IS FAULTY, THEN CHIP E79 IS THE MOST PROBABLE SOURCE OF THE ERROR. IF REPAIRING THAT CHIP DOES NOT REMOVE THE FAULT THEN TRY EACH OF THE CHIPS ASSOCIATED WITH BITS A12, B12 AND D12 SHOULD BE TRIED WITH EQUAL PROBABILITY OF THE FAULT BEING IN ANY ONE OF THESE OTHER THREE CHIPS, TRY CHIPS E61, E86 AND E78.

```
611                                     2.)     IF THERE ARE FOUR CONSECUTIVE BITS IN ERROR,
612                                             FOLLOWING THE PATTERN:
613
614                                             LN, LN+1, LN+2 AND LN+3     WHERE 'L' IS A, B, C OR D
615                                                                         N=0, 4, 8 OR 12
616
617                                             THEN  THE  ABOVE   TABLE   SHOULD   DIRECTLY
618                                             IDENTIFY THE FAILING CHIP.
619
620                                     3.)     IF FOUR  BITS  ARE  DROPPED  WHICH  FIT  THE
621                                             PATTERN:
622
623                                             AN, BN, CN AND DN       WHERE N=15, 14, ... OR 0
624                                             OR 0
625
626                                             THEN ANY ONE OF THE  FOUR  CHIPS  ASSOCIATED
627                                             WITH  EACH  OF  THE  BITS  AN, BN, CN AND DN
628                                             COULD BE AT FAULT WITH EQUAL PROBABILITY.
629
630                                     4.)     IF  16  BITS  ARE  IN  ERROR,  FITTING   THE
631                                             PATTERN:
632
633                                             AN, AN+1, AN+2, AN+3    WHERE N=0, 4, 8 OR 12
634                                             BN, BN+1, BN+2, BN+3
635                                             CN, CN+1, CN+2, CN+3
636                                                     AND
637                                             DN, DN+1, DN+2, AN+3
638
639                                             THEN ANY ONE OF THE  FOUR  CHIPS  ASSOCIATED
640                                             WITH THESE BITS COULD BE AT FAULT WITH EQUAL
641                                             PROBABILITY.
642
643                                     5.)     IF THE FAILING BIT PATTERNS DISPLAYED IN THE
644                                             'AND' AND THE 'OR' DATA TYPED IN THE SUMMARY
645                                             DOES NOT CONFORM EXPLICITELY TO ANY  OF  THE
646                                             ABOVE  PATTERNS,  THEN  THE  TROUBLE SHOOTER
647                                             MUST INTUITIVELY TRY TO FIND  WHICH  OF  THE
648                                             ABOVE CASES (1 THROUGH 4) IS A 'BEST FIT' OF
649                                             THE SYMPTOMS.
650
651                             TEST 12         FPP ACCUMULATORS DUAL ADDRESS TEST
652                             -------         -------------------------------
653
654                             THIS TEST PERFORMS A DUAL  ADDRESSING  TEST  ON  THE
655                             FLOATING  ACCUMULATORS.   NOTE THAT ACCUMULATOR ZERO
656                             IS USED TO ACCESS ALL THE OTHERS.
657
658                             TEST 13         FSRC MODE 0 WITH ILLEGAL ACCUMULATOR TEST
659                             -------         -----------------------------------------
660
661                             THIS IS A TEST OF FSRC MODE 0  WITH  ACCUMULATORS  6
662                             AND   7.    USE   OF   EITHER   OF   THESE   NON-EXISTENT
663                             ACCUMULATORS SHOULD RESULT IN A  TRAP  TO  244  WITH
664                             FEC=2 (ILLEGAL FPP INSTRUCTION).
```

```
                        TEST 14         FSRC MODE 2 TEST
                        -------         ----------------

                                THIS IS A TEST OF FSRC MODE 2, AUTO INCREMENT MODE.

                        TEST 15         FSRC MODE 4 TEST
                        -------         ----------------

                                THIS IS A TEST OF FSRC MODE 4, AUTO DECREMENT MODE.

                        TEST 16         FSRC MODE 2, WITH FD=0, TEST
                        -------         ----------------------------

                                THIS IS A TEST OF FSRC  MODE  2  WITH  FD=0.   (AUTO
                                INCREMENT)

                        TEST 17         FSRC MODE 2 WITH GR7, IMMEDIATE MODE, TEST
                        -------         ------------------------------------------

                                THIS IS A TEST OF FSRC MODE 2 USING  GR7  (THE  PC).
                                THIS IS IMMEDIATE MODE.

                        TEST 20         FSRC MODE 3 TEST
                        -------         ----------------

                                THIS IS A  TEST  OF  FSRC  MODE  3,  AUTO  INCREMENT
                                DEFERRED

                        TEST 21         FSRC MODE 5 TEST
                        -------         ----------------

                                THIS IS A  TEST  Of  FSRC  MODE  5,  AUTO  DECREMENT
                                DEFERRED.

                        TEST 22         FSRC MODE 6 TEST
                        -------         ----------------

                                THIS IS A TEST OF FSRC MODE 6, INDEX MODE

                        TEST 23         FSRC MODE 7 TEST
                        -------         ----------------

                                THIS IS A TEST OF FSRC MODE 7, INDEX DEFERRED MODE.
```

```
                                TEST 24         (BUT EZBT Y8),(BUT ENBT) AND (BUT FIUV) TEST
                                -------         ---------------------------------------------

                                        THIS IS A TEST OF THE (BUT EZBT Y8) FORK,  THE  (BUT
                                        ENBT)   FORK   AND  (BUT  FIUV)  FORK  IN  THE  LOAD
                                        INSTRUCTION FLOWS.
                                        EACH OF THE PATTERNS:

                                                        0
                                                        +NUM
                                                        -NUM
                                                        -0

                                        IS LOADED TWICE, ONCE  WITH  AC>0  THEN  WITH  AC=0.
                                       AFTER  EACH  LOAD  THE  FPS  IS  CHECK TO INSURE THAT
                                        CONTROL WAS PASSED THROUGH WITH THE FORKS PROPERLY.

                                TEST 25         ADDF,ADDD,SUBF AND SUBD WITH FSRC=AC=0 TEST
                                -------         -------------------------------------------

                                        THIS IS A TEST OF ADD AND SUB WITH FSRC=AC=0

                                TEST 26         ADDD AND SUB WITH FSRC=0
                                -------         -----------------------

                                        THIS IS A TEST OF ADD AND SUB WITH FSRC=0.

                                TEST 27         SUBD WITH AC=0 TEST
                                -------         -------------------

                                        THIS IS A TEST OF SUBD WITH AC=0.  BOTH POSITIVE AND
                                        NEGATIVE FSRC'S ARE TRIED.

                                TEST 30         ADDD WITH AC=0 TEST
                                -------         -------------------

                                        POSITIVE AND NEGATIVE FSRC'S ARE TRIED.

                                TEST 31         ADDF AND ADDD WITH E(AC)=E(FSRC) AND (BUT FT) TEST
                                -------         -------------------------------------------------

                                        THIS IS A TEST  OF  THE  ADD  INSTRUCTION  WITH  THE
                                        OPERANDS  HAVING EQUAL EXPONENTS.  THE (BUF FT) FORK
                                        IN THE ROUND/TRUNK FLOWS IS ALSO TESTED.

                                TEST 32         ADDF AND ADDD WITH E(AC) LESS  THAN  E(FSRC) TEST
                                -------         -------------------------------------------------

                                        THIS IS ATEST OF THE ADDD AND ADDF INSTRUCTIONS  AND
                                        THE  ALIGN AC ALGORITHM FLOWS.  THE CONSTANT (25 FOR
                                        FLOATING, 57 FOR  DOUBLE)  USED  IS  CHECKED.   THEN
                                        SIMPLE  AND  WORST  CASE  ALIGNMENT  SITUATIONS  ARE
                                        TRIED.  NOTE E(AC) IS LESS THEN E(FSRC)
```

TEST 33 ADDF AND ADDD WITH E(AC) GREATER THAN E(FSRC) TEST

THIS IS A TEST OF THE ADDD AND ADDF INSTRUCTIONS AND THE ALIGN FSRC ALGORITHM FLOWS. FIRST THE CONSTANT USED IS CHECKED. THEN SIMPLE AND WORST CASE ALIGNMENT SITUATIONS ARE TRIED. NOTE E(AC) IS GREATER THAN E(FSRC).

TEST 34 ADDD WITH NEGATVE OPRANDS TEST

THIS IS A TEST OF THE ADDD INSTRUCTION WITH NEGATIVE OPERANDS. EVERY COMBINATION OF OPERAND SIGNS IS TRIED.

TEST 35 SUBD TEST

THIS IS A TEST OF THE SUBD INSTRUCTION. BOTH A POSITIVE AND A NEGATIVE NUMBER IS SUBTRACTED FROM IT SELF

TEST 36 NORMALIZE ALGORITHM TEST

THIS IS A TEST OF THE NORMALIZE FLOW ALGORITHM. TWO PATTERNS ARE USED, FIRST THE MINIMUM SITUATION REQUIRING ONE LEFT SHIFT AND THEN THE MAXIMUM SITUATION REQUIRING 56 SHIFTS.

```
                                10.                     LISTING
                                                        -------
                                &
                        000213  MNUMBER=213
                        000001  PROGNUM=1
                                        .LIST   ME
                                        .NLIST  MD,MC,CND
000000                          .ENABL  ABS
                                        .MCALL  .HEADER,.SWRHI,.EQUAT,.SETUP,.$CATCH,.$ACT11,.$CMTAG
                                        .MCALL  .$EOP,.$SCOPE,.$ERROR,.$SAVE,.$TYPE,.$TYPOCT
                                        .MCALL  .$TYPDEC,.$TRAP,.$POWER,.$APTHDR,.$APTBLS
                                        .MCALL  .$APTYPE,.$READ
                                        .MCALL  .EQUIV                  ;*REMOVE FOR ASSEMBLY ON PDP-10
                                .TITLE  CKFPAB0 FP11F FLTG PNT PRT A
                                ;*COPYRIGHT (C) 1980
                                ;*DIGITAL EQUIPMENT CORP.
                                ;*MAYNARD, MASS. 01754
                                ;*
                                ;*
                                ;*THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC
                                ;*PACKAGE (MAINDEC-11-DZQAC-C3), JAN 19, 1977.
                                ;*
                        000001  $TN=1
                        160000  $SWR=160000     ;;HALT ON ERROR, LOOP ON TEST, INHIBIT ERROR TYPOUT
                        000244  FPVECT=244
                        177400  $SWR=177400
                        000200  $SWRMSK=200
                        000011  TAB=11
                        000015  CRLF=15
                                .SBTTL  BASIC DEFINITIONS
                                ;*INITIAL ADDRESS OF THE STACK POINTER *** 1100 ***
                        001100  STACK=  1100
                        104000          ERROR=EMT
                        000004          SCOPE=IOT
                                ;*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
                        177776          PSW=PS
                        177774  STKLMT= 177774         ;;STACK LIMIT REGISTER
                        177772  PIRQ=   177772         ;;PROGRAM INTERRUPT REQUEST REGISTER
                        177570  DSWR=   177570         ;;HARDWARE SWITCH REGISTER
                        177570  DDISP=  177570         ;;HARDWARE DISPLAY REGISTER
                                ;*GENERAL PURPOSE REGISTER DEFINITIONS
                        000000  R0=     %0             ;;GENERAL REGISTER
                        000001  R1=     %1             ;;GENERAL REGISTER
                        000002  R2=     %2             ;;GENERAL REGISTER
                        000003  R3=     %3             ;;GENERAL REGISTER
                        000004  R4=     %4             ;;GENERAL REGISTER
                        000005  R5=     %5             ;;GENERAL REGISTER
                        000006  R6=     %6             ;;GENERAL REGISTER
                        000007  R7=     %7             ;;GENERAL REGISTER
                        000006  SP=     %6             ;;STACK POINTER
                        000007  PC=     %7             ;;PROGRAM COUNTER
                                ;*PRIORITY LEVEL DEFINITIONS
```

BASIC 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
001000                          SW9=SW09
000400                          SW8=SW08
000200                          SW7=SW07
000100                          SW6=SW06
000040                          SW5=SW05
000020                          SW4=SW04
000010                          SW3=SW03
000004                          SW2=SW02
000002                          SW1=SW01
000001                          SW0=SW00
                        ;*DATA BIT DEFINITIONS (BIT00 TO BIT15)
100000                  BIT15=  100000
040000                  BIT14=  40000
020000                  BIT13=  20000
010000                  BIT12=  10000
004000                  BIT11=  4000
002000                  BIT10=  2000
001000                  BIT09=  1000
000400                  BIT08=  400
000200                  BIT07=  200
000100                  BIT06=  100
000040                  BIT05=  40
000020                  BIT04=  20
000010                  BIT03=  10
000004                  BIT02=  4
000002                  BIT01=  2
000001                  BIT00=  1
001000                          BIT9=BIT09
000400                          BIT8=BIT08
000200                          BIT7=BIT07
000100                          BIT6=BIT06
000040                          BIT5=BIT05
```

```
                000020                          BIT4=BIT04
                000010                          BIT3=BIT03
                000004                          BIT2=BIT02
                000002                          BIT1=BIT01
                000001                          BIT0=BIT00
                                        ;*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
    978                                 .SBTTL  FPP REGISTER DEFINITIONS
    979         000000                  AC0     =%0
    980         000001                  AC1     =%1
    981         000002                  AC2     =%2
    982         000003                  AC3     =%3
    983         000004                  AC4     =%4
    984         000005                  AC5     =%5
    985         000006                  AC6     =%6
    986         000007                  AC7     =%7
    988                                 .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  003606                  JMP     @#START ;;JUMP TO STARTING ADDRESS OF PROGRAM
```

```
    989                              .SBTTL  COMMON TAGS
                                     ;;*******************************************************************
                                     ;*THIS TABLE CONTAINS VARIOUS COMMON STORAGE LOCATIONS
                                     ;*USED IN THE PROGRAM.
                 001100                      .=1100
        001100                       $CMTAG:                             ;;START OF COMMON TAGS
        001100   000000                      .WORD   0
        001102      000              $TSTNM: .BYTE   0                   ;;CONTAINS THE TEST NUMBER
        001103      000              $ERFLG: .BYTE   0                   ;;CONTAINS ERROR FLAG
        001104   000000              $ICNT:  .WORD   0                   ;;CONTAINS SUBTEST ITERATION COUNT
        001106   000000              $LPADR: .WORD   0                   ;;CONTAINS SCOPE LOOP ADDRESS
        001110   000000              $LPERR: .WORD   0                   ;;CONTAINS SCOPE RETURN FOR ERRORS
        001112   000000              $ERTTL: .WORD   0                   ;;CONTAINS TOTAL ERRORS DETECTED
        001114      000              $ITEMB: .BYTE   0                   ;;CONTAINS ITEM CONTROL BYTE
        001115      001              $ERMAX: .BYTE   1                   ;;CONTAINS MAX. ERRORS PER TEST
        001116   000000              $ERRPC: .WORD   0                   ;;CONTAINS PC OF LAST ERROR INSTRUCTION
        001120   000000              $GDADR: .WORD   0                   ;;CONTAINS ADDRESS OF 'GOOD' DATA
        001122   000000              $BDADR: .WORD   0                   ;;CONTAINS ADDRESS OF 'BAD' DATA
        001124   000000              $GDDAT: .WORD   0                   ;;CONTAINS 'GOOD' DATA
        001126   000000              $BDDAT: .WORD   0                   ;;CONTAINS 'BAD' DATA
        001130   000000                      .WORD   0                   ;;RESERVED--NOT TO BE USED
        001132   000000                      .WORD   0
        001134      000              $AUTOB: .BYTE   0                   ;;AUTOMATIC MODE INDICATOR
        001135      000              $INTAG: .BYTE   0                   ;;INTERRUPT MODE INDICATOR
        001136   000000                      .WORD   0
        001140   177570              SWR:    .WORD   DSWR                ;;ADDRESS OF SWITCH REGISTER
        001142   177570              DISPLAY: .WORD  DDISP               ;;ADDRESS OF DISPLAY REGISTER
        001144   177560              $TKS:   177560                      ;;TTY KBD STATUS
        001146   177562              $TKB:   177562                      ;;TTY KBD BUFFER
        001150   177564              $TPS:   177564                      ;;TTY PRINTER STATUS REG. ADDRESS
        001152   177566              $TPB:   177566                      ;;TTY PRINTER BUFFER REG. ADDRESS
        001154      000              $NULL:  .BYTE   0                   ;;CONTAINS NULL CHARACTER FOR FILLS
        001155      002              $FILLS: .BYTE   2                   ;;CONTAINS # OF FILLER CHARACTERS REQUIRED
        001156      012              $FILLC: .BYTE   12                  ;;INSERT FILL CHARS. AFTER A "LINE FEED"
        001157      000              $TPFLG: .BYTE   0                   ;;"TERMINAL AVAILABLE" FLAG (BIT<07>=0=YES)
        001160   000000              $REGAD: .WORD   0                   ;;CONTAINS THE ADDRESS FROM
                                                                         ;;WHICH  ($REG0) WAS OBTAINED
                 000024                      .REPT   $CM3
        001162   000000              $REG0:  .WORD   0                   ;;CONTAINS (($REGAD)+0)
        001164   000000              $REG1:  .WORD   0                   ;;CONTAINS (($REGAD)+2)
        001166   000000              $REG2:  .WORD   0                   ;;CONTAINS (($REGAD)+4)
        001170   000000              $REG3:  .WORD   0                   ;;CONTAINS (($REGAD)+6)
        001172   000000              $REG4:  .WORD   0                   ;;CONTAINS (($REGAD)+10)
        001174   000000              $REG5:  .WORD   0                   ;;CONTAINS (($REGAD)+12)
        001176   000000              $REG6:  .WORD   0                   ;;CONTAINS (($REGAD)+14)
        001200   000000              $REG7:  .WORD   0                   ;;CONTAINS (($REGAD)+16)
        001202   000000              $REG10: .WORD   0                   ;;CONTAINS (($REGAD)+20)
        001204   000000              $REG11: .WORD   0                   ;;CONTAINS (($REGAD)+22)
        001206   000000              $REG12: .WORD   0                   ;;CONTAINS (($REGAD)+24)
        001210   000000              $REG13: .WORD   0                   ;;CONTAINS (($REGAD)+26)
        001212   000000              $REG14: .WORD   0                   ;;CONTAINS (($REGAD)+30)
        001214   000000              $REG15: .WORD   0                   ;;CONTAINS (($REGAD)+32)
        001216   000000              $REG16: .WORD   0                   ;;CONTAINS (($REGAD)+34)
        001220   000000              $REG17: .WORD   0                   ;;CONTAINS (($REGAD)+36)
        001222   000000              $REG20: .WORD   0                   ;;CONTAINS (($REGAD)+40)
        001224   000000              $REG21: .WORD   0                   ;;CONTAINS (($REGAD)+42)
        001226   000000              $REG22: .WORD   0                   ;;CONTAINS (($REGAD)+44)
```

```
001230  000000                  $REG23: .WORD   0               ;;CONTAINS (($REGAD)+46)
        000024                  .REPT   24
001232  000000                  $TMP0:  .WORD   0               ;;USER DEFINED
001234  000000                  $TMP1:  .WORD   0               ;;USER DEFINED
001236  000000                  $TMP2:  .WORD   0               ;;USER DEFINED
001240  000000                  $TMP3:  .WORD   0               ;;USER DEFINED
001242  000000                  $TMP4:  .WORD   0               ;;USER DEFINED
001244  000000                  $TMP5:  .WORD   0               ;;USER DEFINED
001246  000000                  $TMP6:  .WORD   0               ;;USER DEFINED
001250  000000                  $TMP7:  .WORD   0               ;;USER DEFINED
001252  000000                  $TMP10: .WORD   0               ;;USER DEFINED
001254  000000                  $TMP11: .WORD   0               ;;USER DEFINED
001256  000000                  $TMP12: .WORD   0               ;;USER DEFINED
001260  000000                  $TMP13: .WORD   0               ;;USER DEFINED
001262  000000                  $TMP14: .WORD   0               ;;USER DEFINED
001264  000000                  $TMP15: .WORD   0               ;;USER DEFINED
001266  000000                  $TMP16: .WORD   0               ;;USER DEFINED
001270  000000                  $TMP17: .WORD   0               ;;USER DEFINED
001272  000000                  $TMP20: .WORD   0               ;;USER DEFINED
001274  000000                  $TMP21: .WORD   0               ;;USER DEFINED
001276  000000                  $TMP22: .WORD   0               ;;USER DEFINED
001300  000000                  $TMP23: .WORD   0               ;;USER DEFINED
001302  000000                  $TIMES: 0                       ;;MAX. NUMBER OF ITERATIONS
001304  000000                  $ESCAPE:0                       ;;ESCAPE ON ERROR ADDRESS
001306     207      377     377 $BELL:  .ASCIZ  <207><377><377> ;;CODE FOR BELL
001311     000
001312     077                  $QUES:  .ASCII  /?/             ;;QUESTION MARK
001313     015                  $CRLF:  .ASCII  <15>            ;;CARRIAGE RETURN
001314     012      000         $LF:    .ASCIZ  <12>            ;;LINE FEED
                                ;;*****************************************************************
                                .SBTTL  APT MAILBOX-ETABLE
                                ;;*****************************************************************
                                ;;
                                .EVEN
001316                          $MAIL:                          ;;APT MAILBOX
001316  000000                  $MSGTY: .WORD   AMSGTY          ;;MESSAGE TYPE CODE
001320  000000                  $FATAL: .WORD   AFATAL          ;;FATAL ERROR NUMBER
001322  000000                  $TESTN: .WORD   ATESTN          ;;TEST NUMBER
001324  000000                  $PASS:  .WORD   APASS           ;;PASS COUNT
001326  000000                  $DEVCT: .WORD   ADEVCT          ;;DEVICE COUNT
001330  000000                  $UNIT:  .WORD   AUNIT           ;;I/O UNIT NUMBER
001332  000000                  $MSGAD: .WORD   AMSGAD          ;;MESSAGE ADDRESS
001334  000000                  $MSGLG: .WORD   AMSGLG          ;;MESSAGE LENGTH
001336                          $ETABLE:                        ;;APT ENVIRONMENT TABLE
001336     000                  $ENV:   .BYTE   AENV            ;;ENVIRONMENT BYTE
001337     000                  $ENVM:  .BYTE   AENVM           ;;ENVIRONMENT MODE BITS
001340  000000                  $SWREG: .WORD   ASWREG          ;;APT SWITCH REGISTER
001342  000000                  $USWR:  .WORD   AUSWR           ;;USER SWITCHES
001344  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
001346     000                  $MAMS1: .BYTE   AMAMS1          ;;HIGH ADDRESS,M.S. BYTE
001347     000                  $MTYP1: .BYTE   AMTYP1          ;;MEM. TYPE,BLK#1
                                ;*                              MEM.TYPE BYTE   --  (HIGH BYTE)
```

```
                                        ;*                              900 NSEC CORE=001
                                        ;*                              300 NSEC BIPOLAR=002
                                        ;*                              500 NSEC MOS=003
001350  000000                          $MADR1: .WORD   AMADR1  ;;HIGH ADDRESS,BLK#1
                                        ;*                        MEM.LAST ADDR.=3 BYTES,THIS WORD AND LOW OF ''TYPE'' ABOVE
001352     000                          $MAMS2: .BYTE   AMAMS2  ;;HIGH ADDRESS,M.S. BYTE
001353     000                          $MTYP2: .BYTE   AMTYP2  ;;MEM.TYPE,BLK#2
001354  000000                          $MADR2: .WORD   AMADR2  ;;MEM.LAST ADDRESS,BLK#2
001356     000                          $MAMS3: .BYTE   AMAMS3  ;;HIGH ADDRESS,M.S.BYTE
001357     000                          $MTYP3: .BYTE   AMTYP3  ;;MEM.TYPE,BLK#3
001360  000000                          $MADR3: .WORD   AMADR3  ;;MEM.LAST ADDRESS,BLK#3
001362     000                          $MAMS4: .BYTE   AMAMS4  ;;HIGH ADDRESS,M.S.BYTE
001363     000                          $MTYP4: .BYTE   AMTYP4  ;;MEM.TYPE,BLK#4
001364  000000                          $MADR4: .WORD   AMADR4  ;;MEM.LAST ADDRESS,BLK#4
001366  000000                          $VECT1: .WORD   AVECT1  ;;INTERRUPT VECTOR#1,BUS PRIORITY#1
001370  000000                          $VECT2: .WORD   AVECT2  ;;INTERRUPT VECTOR#2BUS PRIORITY#2
001372  000000                          $BASE:  .WORD   ABASE   ;;BASE ADDRESS OF EQUIPMENT UNDER TEST
001374  000000                          $DEVM:  .WORD   ADEVM   ;;DEVICE MAP
001376  000000                          $CDW1:  .WORD   ACDW1   ;;CONTROLLER DESCRIPTION WORD#1
001400  000000                          $CDW2:  .WORD   ACDW2   ;;CONTROLLER DESCRIPTION WORD#2
001402  000000                          $DDW0:  .WORD   ADDW0   ;;DEVICE DESCRIPTOR WORD#0
001404  000000                          $DDW1:  .WORD   ADDW1   ;;DEVICE DESCRIPTOR WORD#1
001406  000000                          $DDW2:  .WORD   ADDW2   ;;DEVICE DESCRIPTOR WORD#2
001410  000000                          $DDW3:  .WORD   ADDW3   ;;DEVICE DESCRIPTOR WORD#3
001412  000000                          $DDW4:  .WORD   ADDW4   ;;DEVICE DESCRIPTOR WORD#4
001414  000000                          $DDW5:  .WORD   ADDW5   ;;DEVICE DESCRIPTOR WORD#5
001416  000000                          $DDW6:  .WORD   ADDW6   ;;DEVICE DESCRIPTOR WORD#6
001420  000000                          $DDW7:  .WORD   ADDW7   ;;DEVICE DESCRIPTOR WORD#7
001422  000000                          $DDW8:  .WORD   ADDW8   ;;DEVICE DESCRIPTOR WORD#8
001424  000000                          $DDW9:  .WORD   ADDW9   ;;DEVICE DESCRIPTOR WORD#9
001426  000000                          $DDW10: .WORD   ADDW10  ;;DEVICE DESCRIPTOR WORD#10
001430  000000                          $DDW11: .WORD   ADDW11  ;;DEVICE DESCRIPTOR WORD#11
001432  000000                          $DDW12: .WORD   ADDW12  ;;DEVICE DESCRIPTOR WORD#12
001434  000000                          $DDW13: .WORD   ADDW13  ;;DEVICE DESCRIPTOR WORD#13
001436  000000                          $DDW14: .WORD   ADDW14  ;;DEVICE DESCRIPTOR WORD#14
001440  000000                          $DDW15: .WORD   ADDW15  ;;DEVICE DESCRIPTOR WORD#15
001442                                  $ETEND:
```

ERROR POINTER TABLE

```
                                        .SBTTL  ERROR POINTER TABLE
                                        ;*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
                                        ;
                                        $ERRTB:
    001442
996 001442  043760  064203  070616          .WORD   EM1,DH1,DT1,DF1         ;ERROR ITEM # 1
    001452  044015  064273  070640          .WORD   EM2,DH2,DT2,DF2         ;ERROR ITEM # 2
    001462  044061  064366  070662          .WORD   EM3,DH3,DT3,DF3         ;ERROR ITEM # 3
    001472  044126  064457  070662          .WORD   EM4,DH4,DT4,DF4         ;ERROR ITEM # 4
    001502  044166  064553  070704          .WORD   EM5,DH5,DT5,DF5         ;ERROR ITEM # 5
    001512  044222  064553  070736          .WORD   EM6,DH6,DT6,DF6         ;ERROR ITEM # 6
    001522  044254  064553  070736          .WORD   EM7,DH7,DT7,DF7         ;ERROR ITEM # 7
    001532  044166  064553  070736          .WORD   EM10,DH10,DT10,DF10     ;ERROR ITEM # 10
    001542  044307  064553  070736          .WORD   EM11,DH11,DT11,DF11     ;ERROR ITEM # 11
    001552  000000  000000  070760          .WORD   EM12,DH12,DT12,DF12     ;ERROR ITEM # 12
    001562  000000  000000  071046          .WORD   EM13,DH13,DT13,DF13     ;ERROR ITEM # 13
    001572  044370  064553  070736          .WORD   EM14,DH14,DT14,DF14     ;ERROR ITEM # 14
    001602  044513  064553  070736          .WORD   EM15,DH15,DT15,DF15     ;ERROR ITEM # 15
    001612  044636  064613  071100          .WORD   EM16,DH16,DT16,DF16     ;ERROR ITEM # 16
    001622  044707  064673  070662          .WORD   EM17,DH17,DT17,DF17     ;ERROR ITEM # 17
    001632  045142  064763  071120          .WORD   EM20,DH20,DT20,DF20     ;ERROR ITEM # 20
    001642  045320  064553  071142          .WORD   EM21,DH21,DT21,DF21     ;ERROR ITEM # 21
    001652  045451  065051  071154          .WORD   EM22,DH22,DT22,DF22     ;ERROR ITEM # 22
    001662  045451  065106  071202          .WORD   EM23,DH23,DT23,DF23     ;ERROR ITEM # 23
    001672  045451  065244  071224          .WORD   EM24,DH24,DT24,DF24     ;ERROR ITEM # 24
    001702  045536  065403  071250          .WORD   EM25,DH25,DT25,DF25     ;ERROR ITEM # 25
    001712  045651  065445  071320          .WORD   EM26,DH26,DT26,DF26     ;ERROR ITEM # 26
    001722  045651  065445  071374          .WORD   EM27,DH27,DT27,DF27     ;ERROR ITEM # 27
    001732  045717  000000  071436          .WORD   EM30,DH30,DT30,DF30     ;ERROR ITEM # 30
    001742  045771  065445  071320          .WORD   EM31,DH31,DT31,DF31     ;ERROR ITEM # 31
    001752  045771  065445  071374          .WORD   EM32,DH32,DT32,DF32     ;ERROR ITEM # 32
    001762  046037  065533  071470          .WORD   EM33,DH33,DT33,DF33     ;ERROR ITEM # 33
    001772  046100  065533  071546          .WORD   EM34,DH34,DT34,DF34     ;ERROR ITEM # 34
    002002  046202  065533  071546          .WORD   EM35,DH35,DT35,DF35     ;ERROR ITEM # 35
    002012  046304  065533  071546          .WORD   EM36,DH36,DT36,DF36     ;ERROR ITEM # 36
    002022  046405  065533  071546          .WORD   EM37,DH37,DT37,DF37     ;ERROR ITEM # 37
    002032  046506  065533  071470          .WORD   EM40,DH40,DT40,DF40     ;ERROR ITEM # 40
    002042  046657  000000  071620          .WORD   EM41,DH41,DT41,DF41     ;ERROR ITEM # 41
    002052  046714  065636  071652          .WORD   EM42,DH42,DT42,DF42     ;ERROR ITEM # 42
    002062  047035  065636  071652          .WORD   EM43,DH43,DT43,DF43     ;ERROR ITEM # 43
    002072  047156  000000  071730          .WORD   EM44,DH44,DT44,DF44     ;ERROR ITEM # 44
    002102  047156  065740  072000          .WORD   EM45,DH45,DT45,DF45     ;ERROR ITEM # 45
    002112  047221  065757  072054          .WORD   EM46,DH46,DT46,DF46     ;ERROR ITEM # 46
    002122  047277  065740  072142          .WORD   EM47,DH47,DT47,DF47     ;ERROR ITEM # 47
    002132  047415  066003  071546          .WORD   EM50,DH50,DT50,DF50     ;ERROR ITEM # 50
    002142  047513  066003  072174          .WORD   EM51,DH51,DT51,DF51     ;ERROR ITEM # 51
    002152  047554  064553  072142          .WORD   EM52,DH52,DT52,DF52     ;ERROR ITEM # 52
    002162  047675  065445  072232          .WORD   EM53,DH53,DT53,DF53     ;ERROR ITEM # 53
    002172  050072  066055  072252          .WORD   EM54,DH54,DT54,DF54     ;ERROR ITEM # 54
    002202  050136  064553  072142          .WORD   EM55,DH55,DT55,DF55     ;ERROR ITEM # 55
    002212  050257  065445  072232          .WORD   EM56,DH56,DT56,DF56     ;ERROR ITEM # 56
```

ERROR POINTER TABLE

```
002222  050454  066055  072252          .WORD   EM57,DH57,DT57,DF57          ;ERROR ITEM # 57
002232  050520  065445  072232          .WORD   EM60,DH60,DT60,DF60          ;ERROR ITEM # 60
002242  050715  066055  072252          .WORD   EM61,DH61,DT61,DF61          ;ERROR ITEM # 61
002252  050761  066055  072252          .WORD   EM62,DH62,DT62,DF62          ;ERROR ITEM # 62
002262  051153  066055  072252          .WORD   EM63,DH63,DT63,DF63          ;ERROR ITEM # 63
002272  051345  066165  072310          .WORD   EM64,DH64,DT64,DF64          ;ERROR ITEM # 64
002302  051345  066116  072310          .WORD   EM65,DH65,DT65,DF65          ;ERROR ITEM # 65
002312  051501  066055  072252          .WORD   EM66,DH66,DT66,DF66          ;ERROR ITEM # 66
002322  051544  064553  071142          .WORD   EM67,DH67,DT67,DF67          ;ERROR ITEM # 67
002332  051775  064553  072330          .WORD   EM70,DH70,DT70,DF70          ;ERROR ITEM # 70
002342  052120  065403  072330          .WORD   EM71,DH71,DT71,DF71          ;ERROR ITEM # 71
002352  052222  065445  072376          .WORD   EM72,DH72,DT72,DF72          ;ERROR ITEM # 72
002362  052276  066055  072252          .WORD   EM73,DH73,DT73,DF73          ;ERROR ITEM # 73
002372  052336  064553  071142          .WORD   EM74,DH74,DT74,DF74          ;ERROR ITEM # 74
002402  052567  064553  072330          .WORD   EM75,DH75,DT75,DF75          ;ERROR ITEM # 75
002412  052712  065403  072330          .WORD   EM76,DH76,DT76,DF76          ;ERROR ITEM # 76
002422  053014  065445  072376          .WORD   EM77,DH77,DT77,DF77          ;ERROR ITEM # 77
002432  053070  066055  072252          .WORD   EM100,DH100,DT100,DF100          ;ERROR ITEM # 100
002442  053130  064553  072330          .WORD   EM101,DH101,DT101,DF101          ;ERROR ITEM # 101
002452  053254  065445  072330          .WORD   EM102,DH102,DT102,DF102          ;ERROR ITEM # 102
002462  053326  065403  072330          .WORD   EM103,DH103,DT103,DF103          ;ERROR ITEM # 103
002472  053431  066055  072252          .WORD   EM104,DH104,DT104,DF104          ;ERROR ITEM # 104
002502  053472  064553  072330          .WORD   EM105,DH105,DT105,DF105          ;ERROR ITEM # 105
002512  053617  065445  072376          .WORD   EM106,DH106,DT106,DF106          ;ERROR ITEM # 106
002522  053672  065403  072330          .WORD   EM107,DH107,DT107,DF107          ;ERROR ITEM # 107
002532  053776  066055  072252          .WORD   EM110,DH110,DT110,DF110          ;ERROR ITEM # 110
002542  054040  065403  072416          .WORD   EM111,DH111,DT111,DF111          ;ERROR ITEM # 111
002552  054040  066253  072416          .WORD   EM112,DH112,DT112,DF112          ;ERROR ITEM # 112
002562  054142  065403  072416          .WORD   EM113,DH113,DT113,DF113          ;ERROR ITEM # 113
002572  054142  066253  072416          .WORD   EM114,DH114,DT114,DF114          ;ERROR ITEM # 114
002602  054040  066472  072416          .WORD   EM115,DH115,DT115,DF115          ;ERROR ITEM # 115
002612  054142  066472  072416          .WORD   EM116,DH116,DT116,DF116          ;ERROR ITEM # 116
002622  054244  064673  070662          .WORD   EM117,DH117,DT117,DF117          ;ERROR ITEM # 117
002632  054400  066756  070662          .WORD   EM120,DH120,DT120,DF120          ;ERROR ITEM # 120
002642  054534  064553  072142          .WORD   EM121,DH121,DT121,DF121          ;ERROR ITEM # 121
002652  054653  066003  071546          .WORD   EM122,DH122,DT122,DF122          ;ERROR ITEM # 122
002662  054752  066003  072174          .WORD   EM123,DH123,DT123,DF123          ;ERROR ITEM # 123
002672  055013  064673  072430          .WORD   EM124,DH124,DT124,DF124          ;ERROR ITEM # 124
002702  055106  064673  072430          .WORD   EM125,DH125,DT125,DF125          ;ERROR ITEM # 125
002712  055176  064553  072416          .WORD   EM126,DH126,DT126,DF126          ;ERROR ITEM # 126
002722  055405  066055  072416          .WORD   EM127,DH127,DT127,DF127          ;ERROR ITEM # 127
002732  055620  066756  070662          .WORD   EM130,DH130,DT130,DF130          ;ERROR ITEM # 130
002742  055720  066055  072514          .WORD   EM131,DH131,DT131,DF131          ;ERROR ITEM # 131
002752  055760  066055  072514          .WORD   EM132,DH132,DT132,DF132          ;ERROR ITEM # 132
002762  056020  067046  072556          .WORD   EM133,DH133,DT133,DF133          ;ERROR ITEM # 133
002772  056057  067046  072556          .WORD   EM134,DH134,DT134,DF134          ;ERROR ITEM # 134
003002  056116  067046  072556          .WORD   EM135,DH135,DT135,DF135          ;ERROR ITEM # 135
003012  056155  067046  072556          .WORD   EM136,DH136,DT136,DF136          ;ERROR ITEM # 136
003022  056020  067156  072630          .WORD   EM137,DH137,DT137,DF137          ;ERROR ITEM # 137
003032  056057  067156  072630          .WORD   EM140,DH140,DT140,DF140          ;ERROR ITEM # 140
003042  056116  067156  072630          .WORD   EM141,DH141,DT141,DF141          ;ERROR ITEM # 141
003052  056155  067156  072630          .WORD   EM142,DH142,DT142,DF142          ;ERROR ITEM # 142
003062  056214  067046  072556          .WORD   EM143,DH143,DT143,DF143          ;ERROR ITEM # 143
003072  056247  067046  072556          .WORD   EM144,DH144,DT144,DF144          ;ERROR ITEM # 144
003102  056214  067156  072630          .WORD   EM145,DH145,DT145,DF145          ;ERROR ITEM # 145
003112  056247  067156  072630          .WORD   EM146,DH146,DT146,DF146          ;ERROR ITEM # 146
003122  056302  066055  072556          .WORD   EM147,DH147,DT147,DF147          ;ERROR ITEM # 147
```

ERROR POINTER TABLE

```
003132  056302  067346  072556          .WORD   EM150,DH150,DT150,DF150          ;ERROR ITEM # 150
003142  056302  067156  072630          .WORD   EM151,DH151,DT151,DF151          ;ERROR ITEM # 151
003152  056334  067046  072556          .WORD   EM152,DH152,DT152,DF152          ;ERROR ITEM # 152
003162  056334  067156  072630          .WORD   EM153,DH153,DT153,DF153          ;ERROR ITEM # 153
003172  056366  067437  072650          .WORD   EM154,DH154,DT154,DF154          ;ERROR ITEM # 154
003202  056620  067437  072650          .WORD   EM155,DH155,DT155,DF155          ;ERROR ITEM # 155
003212  057053  066055  072556          .WORD   EM156,DH156,DT156,DF156          ;ERROR ITEM # 156
003222  057270  066055  072556          .WORD   EM157,DH157,DT157,DF157          ;ERROR ITEM # 157
003232  057507  066055  072556          .WORD   EM160,DH160,DT160,DF160          ;ERROR ITEM # 160
003242  057714  066055  072556          .WORD   EM161,DH161,DT161,DF161          ;ERROR ITEM # 161
003252  060121  066055  072556          .WORD   EM162,DH162,DT162,DF162          ;ERROR ITEM # 162
003262  060166  066055  072556          .WORD   EM163,DH163,DT163,DF163          ;ERROR ITEM # 163
003272  060233  064673  070662          .WORD   EM164,DH164,DT164,DF164          ;ERROR ITEM # 164
003302  060300  064673  070662          .WORD   EM165,DH165,DT165,DF165          ;ERROR ITEM # 165
003312  060345  066055  072556          .WORD   EM166,DH166,DT166,DF166          ;ERROR ITEM # 166
003322  060455  066055  072556          .WORD   EM167,DH167,DT167,DF167          ;ERROR ITEM # 167
003332  060714  066055  072556          .WORD   EM170,DH170,DT170,DF170          ;ERROR ITEM # 170
003342  061024  066055  072556          .WORD   EM171,DH171,DT171,DF171          ;ERROR ITEM # 171
003352  061263  066055  072556          .WORD   EM172,DH172,DT172,DF172          ;ERROR ITEM # 172
003362  061522  066055  072556          .WORD   EM173,DH173,DT173,DF173          ;ERROR ITEM # 173
003372  061761  066055  072556          .WORD   EM174,DH174,DT174,DF174          ;ERROR ITEM # 174
003402  062220  066055  072556          .WORD   EM175,DH175,DT175,DF175          ;ERROR ITEM # 175
003412  062457  066055  072556          .WORD   EM176,DH176,DT176,DF176          ;ERROR ITEM # 176
003422  062614  066055  072556          .WORD   EM177,DH177,DT177,DF177          ;ERROR ITEM # 177
003432  062751  066055  072556          .WORD   EM200,DH200,DT200,DF200          ;ERROR ITEM # 200
003442  063106  066055  072556          .WORD   EM201,DH201,DT201,DF201          ;ERROR ITEM # 201
003452  063243  066055  072556          .WORD   EM202,DH202,DT202,DF202          ;ERROR ITEM # 202
003462  063400  066055  072556          .WORD   EM203,DH203,DT203,DF203          ;ERROR ITEM # 203
003472  063535  066055  072556          .WORD   EM204,DH204,DT204,DF204          ;ERROR ITEM # 204
003502  063672  064673  070662          .WORD   EM205,DH205,DT205,DF205          ;ERROR ITEM # 205
003512  063737  066055  072556          .WORD   EM206,DH206,DT206,DF206          ;ERROR ITEM # 206
003522  064004  066055  072556          .WORD   EM207,DH207,DT207,DF207          ;ERROR ITEM # 207
003532  064126  066055  072556          .WORD   EM210,DH210,DT210,DF210          ;ERROR ITEM # 210
003542  044166  067477  072662          .WORD   EM211,DH211,DT211,DF211          ;ERROR ITEM # 211
003552  044222  064553  072700          .WORD   EM212,DH212,DT212,DF212          ;ERROR ITEM # 212
003562  044254  064553  072700          .WORD   EM213,DH213,DT213,DF213          ;ERROR ITEM # 213
```

```
1000                                .SBTTL  ACT11 HOOKS
                                    ;;*****************************************************************
                                    ;HOOKS REQUIRED BY ACT11
            003572                          $SVPC=.                 ;SAVE PC
            000046                          .=46
      000046 035436                         $ENDAD                  ;;1)SET LOC.46 TO ADDRESS OF $ENDAD IN .$EOP
            000052                          .=52
      000052 000000                         .WORD   0               ;;2)SET LOC.52 TO ZERO
            003572                          .=$SVPC                 ;; RESTORE PC
```

```
1002                                    .SBTTL  APT PARAMETER BLOCK
                                        ;;**********************************************************************
                                        ;SET LOCATIONS 24 AND 44 AS REQUIRED FOR APT
                                        ;;**********************************************************************
            003572                              .$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 003572                             $APTHDR  ;;POINT TO APT HEADER BLOCK
            003572                              .=.$X    ;;RESET LOCATION COUNTER
                                        ;;**********************************************************************
                                        ;SETUP APT PARAMETER BLOCK AS DEFINED IN THE APT-PDP11 DIAGNOSTIC
                                        ;INTERFACE SPEC.
      003572                            $APTHD:
      003572 000000                     $HIBTS: .WORD   0          ;;TWO HIGH BITS OF 18 BIT MAILBOX ADDR.
      003574 001316                     $MBADR: .WORD   $MAIL      ;;ADDRESS OF APT MAILBOX (BITS 0-15)
      003576 000010                     $TSTM:  .WORD   10         ;;RUN TIM OF LONGEST TEST
      003600 000040                     $PASTM: .WORD   40         ;;RUN TIME IN SECS. OF 1ST PASS ON 1 UNIT (QUICK VERIFY)
      003602 000000                     $UNITM: .WORD   0          ;;ADDITIONAL RUN TIME (SECS) OF A PASS FOR EACH ADDITIONAL UNIT
      003604 000052                             .WORD   $ETEND-$MAIL/2 ;;LENGTH MAILBOX-ETABLE(WORDS)
```

```
1004                                    .SBTTL  INITIALIZE THE COMMON TAGS
1005 003606                             START:
                                        .SBTTL  INITIALIZE THE COMMON TAGS
                                        ;;CLEAR THE COMMON TAGS ($CMTAG) AREA
003606  012706  001100                          MOV     #$CMTAG,R6      ;;FIRST LOCATION TO BE CLEARED
003612  005026                                  CLR     (R6)+           ;;CLEAR MEMORY LOCATION
003614  022706  001140                          CMP     #SWR,R6 ;;DONE?
003620  001374                                  BNE     .-6             ;;LOOP BACK IF NO
003622  012706  001100                          MOV     #STACK,SP       ;;SETUP THE STACK POINTER
                                        ;;INITIALIZE A FEW VECTORS
003626  012737  035516  000020                  MOV     #$SCOPE,@#IOTVEC ;;IOT VECTOR FOR SCOPE ROUTINE
003634  012737  000340  000022                  MOV     #340,@#IOTVEC+2 ;;LEVEL 7
003642  012737  035776  000030                  MOV     #$ERROR,@#EMTVEC ;;EMT VECTOR FOR ERROR ROUTINE
003650  012737  000340  000032                  MOV     #340,@#EMTVEC+2 ;;LEVEL 7
003656  012737  040240  000034                  MOV     #$TRAP,@#TRAPVEC ;;TRAP VECTOR FOR TRAP CALLS
003664  012737  000340  000036                  MOV     #340,@#TRAPVEC+2;LEVEL 7
003672  012737  040324  000024                  MOV     #$PWRDN,@#PWRVEC ;;POWER FAILURE VECTOR
003700  012737  000340  000026                  MOV     #340,@#PWRVEC+2 ;;LEVEL 7
003706  013737  035256  035250                  MOV     $ENDCT,$EOPCT   ;;SETUP END-OF-PROGRAM COUNTER
003714  005037  001302                          CLR     $TIMES          ;;INITIALIZE NUMBER OF ITERATIONS
003720  005037  001304                          CLR     $ESCAPE         ;;CLEAR THE ESCAPE ON ERROR ADDRESS
003724  112737  000001  001115                  MOVB    #1,$ERMAX       ;;ALLOW ONE ERROR PER TEST
                                        ;;INITIALIZE THE "T-BIT" TRAP VECTOR. THEN LOAD LOCATION "$RTRN", IN
                                        ;;THE "END-OF-PASS" ($EOP) ROUTINE, WITH A "RTI" OR "RTT".
003732  012737  035502  000014                  MOV     #$RTRN,@#TBITVEC ;;SET "T" BIT VECTOR TO $RTRN
003740  012737  000340  000016                  MOV     #340,@#TBITVEC+2 ;;LEVEL 7
003746  012737  000002  035502                  MOV     #RTI,$RTRN      ;;SET $RTRN TO A RTI
003754  012737  004002  000010                  MOV     #65$,@#RESVEC   ;;TRY TO DO A RTT
003762  005046                                  CLR     -(SP)           ;;DUMMY PS
003764  012746  003772                          MOV     #64$,-(SP)      ;;AND PC
003770  000006                                  RTT                     ;;TRY THE RTT
003772  012737  000006  035502  64$:     MOV     #RTT,$RTRN      ;;RTT IS LEGAL--SET $RTRN TO A RTT
004000  000402                                  BR      66$
004002  062706  000010          65$:     ADD     #10,SP          ;;RTT ILLEGAL--CLEAN OFF THE STACK
004006  012737  000012  000010  66$:     MOV     #RESVEC+2,@#RESVEC ;;RESTORE TRAP CATCHER
004014  005037  035510                          CLR     $TBIT           ;;CLEAR "T" BIT SWITCH
004020  012737  004020  001106                  MOV     #.,$LPADR       ;;INITIALIZE THE LOOP ADDRESS FOR SCOPE
004026  012737  004026  001110                  MOV     #.,$LPERR       ;;SETUP THE ERROR LOOP ADDRESS
                                        ;;SIZE FOR A HARDWARE SWITCH REGISTER. IF NOT FOUND OR IT IS
                                        ;;EQUAL TO A "-1", SETUP FOR A SOFTWARE SWITCH REGISTER.
004034  013746  000004                          MOV     @#ERRVEC,-(SP)  ;;SAVE ERROR VECTOR
004040  012737  004074  000004                  MOV     #67$,@#ERRVEC   ;;SET UP ERROR VECTOR
004046  012737  177570  001140                  MOV     #DSWR,SWR       ;;SETUP FOR A HARDWARE SWICH REGISTER
004054  012737  177570  001142                  MOV     #DDISP,DISPLAY  ;;AND A HARDWARE DISPLAY REGISTER
004062  022777  177777  175050                  CMP     #-1,@SWR        ;;TRY TO REFERENCE HARDWARE SWR
004070  001012                                  BNE     69$             ;;BRANCH IF NO TIMEOUT TRAP OCCURRED
                                                                        ;;AND  THE HARDWARE SWR IS NOT = -1
004072  000403                                  BR      68$             ;;BRANCH IF NO TIMEOUT
004074  012716  004102          67$:     MOV     #68$,(SP)       ;;SET UP FOR TRAP RETURN
004100  000002                                  RTI
004102  012737  000176  001140  68$:     MOV     #SWREG,SWR      ;;POINT TO SOFTWARE SWR
004110  012737  000174  001142                  MOV     #DISPREG,DISPLAY
004116  012637  000004          69$:     MOV     (SP)+,@#ERRVEC  ;;RESTORE ERROR VECTOR
004122  005037  001324                          CLR     $PASS           ;;CLEAR PASS COUNT
004126  132737  000200  001337                  BITB    #APTSIZE,$ENVM  ;;TEST USER SIZE UNDER APT
004134  001403                                  BEQ     70$             ;;YES,USE NON-APT SWITCH
004136  012737  001340  001140                  MOV     #$SWREG,SWR     ;;NO,USE APT SWITCH REGISTER
```

INITIALIZE THE COMMON TAGS

```
      004144                                  70$:
1006                                          .SBTTL  TYPE PROGRAM NAME
                                              ;;TYPE THE NAME OF THE PROGRAM IF FIRST PASS
      004144  005227  177777                   INC     #-1             ;;FIRST TIME?
      004150  001047                          BNE     71$             ;;BRANCH IF NO
      004152  022737  035436  000042          CMP     #$ENDAD,@#42    ;;ACT-11?
      004160  001443                          BEQ     71$             ;;BRANCH IF YES
      004162  104401  004230                  TYPE    ,72$            ;;TYPE ASCIZ STRING
                                              .SBTTL  GET VALUE FOR SOFTWARE SWITCH REGISTER
      004166  005737  000042                   TST     @#42            ;;ARE WE RUNNING UNDER XXDP/ACT?
      004172  001012                          BNE     73$             ;;BRANCH IF YES
      004174  123727  001336  000001          CMPB    $ENV,#1         ;;ARE WE RUNNING UNDER APT?
      004202  001406                          BEQ     73$             ;;BRANCH IF YES
      004204  023727  001140  000176          CMP     SWR,#SWREG      ;;SOFTWARE SWITCH REG SELECTED?
      004212  001005                          BNE     74$             ;;BRANCH IF NO
      004214  104406                          GTSWR                   ;;GET SOFT-SWR SETTINGS
      004216  000403                          BR      74$
      004220  112737  000001  001134  73$:    MOVB    #1,$AUTOB       ;;SET AUTO-MODE INDICATOR
      004226                                  74$:
      004226  000420                          BR      71$             ;;GET OVER THE ASCIZ
                                              ;;72$:  .ASCIZ  <CRLF>*CKFPABO FP11F FLTG PNT PRT A*<CRLF>
      004270                                  71$:
1007  004270                                  LOOP:
```

```
                                .SBTTL  TEST # 1 - LDFPS, STFPS AND DATA PATHS TEST
                                ;;*****************************************************************
                                ;*TEST 1 - LDFPS, STFPS AND DATA PATHS TEST
                                ;*
                                ;*THIS IS A TEST OF THE LDFPS (LOAD FLOATING POINT STATUS) AND STFPS
                                ;*(STORE FLOATING POINT STATUS) INSTRUCTIONS. A COUNT PATTERN IS GENERATED
                                ;*AND RUN THROUGH THE FLOATING POINT STATUS REGISTER.
                                ;*THIS WILL TEST THE 16-BIT TRI STATE BUS WHICH CONNECTS THE CPU
                                ;*WITH THE FPP AND ALSO RUNS INTERNALLY WITHIN THE FPP. ONLY DMO AND
                                ;*SMO ARE USED.
                                ;*NOTE THAT A MASK MUST BE USED BECAUSE SOME OF THE FPS BITS CANNOT
                                ;*BE SET.
                                ;*
                                ;*ONLY THE FIRST FIVE ERRORS WILL BE REPORTED INDIVIDUALLY.
                                ;*THIS IS TO PREVENT LOCKING OUT THE COMPLETION OF THE TEST BECAUSE
                                ;*OF VIRTUALLY ENDLESS NUMBER OF ERRORS. ONLY FIVE INDIVIDUAL ERRORS
                                ;*WILL BE REPORTED THEN THE TEST WILL BE COMPLETED AND AN ERROR
                                ;*SUMMARY GIVEN (SEE NOTE BELOW).
                                ;*
                                ;*NOTE THAT THIS TEST KEEPS A DYNAMIC RECORD OF THE LOGICAL 'AND' AND 'OR'
                                ;*OF THE FAILING DATA PATTERNS. THESE CAN BE VERY USEFUL IN DETERMINING
                                ;*STUCK BITS. IF THE USER HAS THE INHIBIT ERROR TYPE OUT SWITCH (SWR13)
                                ;*OFF, THEN THE USER WILL RECIEVE EACH INDIVIDUAL ERROR MESSAGE PLUS
                                ;*AN ERROR SUMMARY AT THE END OF THE TEST. INHIBITING ERROR PRINT OUT
                                ;*WILL INHIBIT ERROR SUMMARY PRINT OUT, EXCEPT IN THE CASE DESCRIBED BELOW.
                                ;*TO GET JUST THE ERROR SUMMARY WITH NO INDIVIDUAL ERROR REPORTS,
                                ;*SET SWITCH REGISTER BIT13 AND SWITCH REGISTER BIT7 BOTH ON.
                                ;*
                                ;*****************************************************************
     004270  000004             TST1:   SCOPE
1027 004272  005037  004550             CLR     AERFLG
1028 004276  012737  004340  001110     MOV     #A1,$LPERR      ;SET UP THE LOOP ON ERROR ADDRESS.
1029 004304  012700  177777             MOV     #-1,R0          ;INITIALIZE THE COUNT PATTERN.
1030 004310  012737  004552  000244     MOV     #AERR1,FPVECT   ;SET UP FOR UNABLE TO DECODE
1031 004316  012737  004564  000010     MOV     #AERR2,10       ;FPP INSTRUCTION TRAP TO 244 OR 10.
1032 004324  005002                     CLR     R2              ;R2 IS THE 'AND' OF BAD DATA.
1033 004326  005102                     COM     R2
1034 004330  005003                     CLR     R3              ;R3 IS THE 'OR' OF BAD DATA.
1035 004332  012737  004616  000004     MOV     #AERR3,ERRVECT  ;IF EITHER INSTRUCTION
1036                                                             ;FAILS TO GO THROUGH THE
1037                                                             ;CORRECT SRC OR DST MODE AN
1038                                                             ;ODD ADDRESS TRAP WILL OCCUR.
1039 004340                     A1:
1040 004340  010004             A11:    MOV     R0,R4
1041 004342  042704  030020             BIC     #30020,R4
1042 004346  170104                     LDFPS   R4              ;TEST INSTRUCTION.
1043
1044 004350  012701  177777             MOV     #-1,R1
1045 004354  170201             A12:    STFPS   R1              ;TEST INSTRUCTION.
1046 004356  012737  041142  000244     MOV     #FPSPUR,FPVECT  ;SET UP FOR UNEXPECTED TRAPS.
1047 004364  010004                     MOV     R0,R4           ;MASK OFF UNSETTABLE BITS.
1048 004366  042704  030020             BIC     #30020,R4
1049 004372  012737  041174  000004     MOV     #CPSPUR,ERRVECT
1050 004400  012737  041212  000010     MOV     #CPTWO,10
1051 004406  020401                     CMP     R4,R1           ;COMPARE DATA EXPECTED WITH
1052                                                             ;THE DATA READ.
1053 004410  001002                     BNE     A3              ;IF NOT EQUAL GO REPORT ERROR.
```

```
1054
1055 004412  077026                    A2:     SOB     R0,A1                   ;OTHERWISE DECREMENT COUNT PATTERN
1056 004414  000425                            BR      A5                      ;UNTIL IT IS ZERO.
1057
1058 004416  005237  004550              A3:     INC     AERFLG                  ;RECORD ERROR.
1059 004422  050003                            BIS     R0,R3                   ;COMPUTE 'OR' OF FAILING PATTERNS.
1060 004424  010005                            MOV     R0,R5                   ;COMPUTE 'AND' OF FAILING PATTERNS.
1061 004426  005105                            COM     R5
1062 004430  040502                            BIC     R5,R2
1063
1064 004432  022737  000005  004550            CMP     #5,AERFLG               ;SEE IF MORE THAN 5 ERRORS HAVE
1065 004440  103412                            BLO     A05                     ;OCCURRED. BR IF YES.
1066                                                                           ;OTHERWISE
1067                                                                           ;REPORT ERROR.
1068 004442  012737  004340  001236            MOV     #A1,$TMP2
1069 004450  010037  001240                    MOV     R0,$TMP3
1070 004454  010137  001242                    MOV     R1,$TMP4
1071 004460  010437  001244                    MOV     R4,$TMP5
1072 004464  104001                    A4:     ERROR   +1
1073
1074 004466  000751                    A05:    BR      A2                      ;CONTINUE TESTING.
1075
1076 004470  005737  004550              A5:     TST     AERFLG                  ;SEE IF ANY ERRORS OCCURRED.
1077 004474  001471                            BEQ     ADONE                   ;IF NOT GO TO NEXT TEST.
1078 004476  032777  020000  174434            BIT     #SW13,@SWR              ;OTHERWISE SEE IF A SUMMARY
1079 004504  001404                            BEQ     A6                      ;SHOULD BE TYPED.
1080 004506  032777  000200  174424            BIT     #SW7,@SWR
1081 004514  001461                            BEQ     ADONE
1082
1083 004516                             A6:                                     ;TYPE ERROR SUMMARY.
1084 004516  010237  001236                    MOV     R2,$TMP2
1085 004522  010337  001240                    MOV     R3,$TMP3
1086 004526  012737  004542  001116            MOV     #A7,$ERRPC
1087 004534  112737  000002  001114            MOVB    #2,$ITEMB
1088 004542  004737  040510              A7:     JSR     PC,ERTYPE
1089 004546  000444                            BR      ADONE
1090
1091 004550  000000                    AERFLG:.WORD    0
1092
1093                                    ;UNABLE TO DECODE FPP INSTRUCTION. TRAPPED TO 244.
1094 004552  011637  001236            AERR1:  MOV     (SP),$TMP2              ;SAVE PC OF TRAP.
1095 004556  022626                            CMP     (SP)+,(SP)+
1096 004560  104010                    1$:     ERROR   +10
1097 004562  000436                            BR      ADONE
1098
1099                                    ;UNABLE TO DECODE INSTRUCTION. TRAPPED TO 10.
1100 004564  021627  004342            AERR2:  CMP     (SP),#A11+2             ;DID TRAP OCCUR OF FPP INSTRUCTION?
1101 004570  001405                            BEQ     1$
1102 004572  021627  004356                    CMP     (SP),#A12+2
1103 004576  001402                            BEQ     1$
1104 004600  000137  041212                    JMP     CPTWO                   ;IF NOT FPP INSTRUCTION THEN
1105                                                                           ;REPORT SPURIOUS TRAP TO 10.
1106
1107 004604  011637  001236            1$:     MOV     (SP),$TMP2              ;OTHERWISE REPORT IR DECIDE ERROR.
1108 004610  022626                            CMP     (SP)+,(SP)+
1109 004612  104011                    2$:     ERROR   +11
1110 004614  000421                            BR      ADONE
```

```
1111
1112                                    ;TRAP TO 4 HANDLER:
1113 004616  021627  004342    AERR3:   CMP     (SP),#A11+2          ;DID THE TRAP OCCUR ON THE
1114 004622  001405                     BEQ     1$                   ;LDFPS INSTRUCTION?
1115 004624  021627  004356             CMP     (SP),#A12+2          ;OR THE STFPS INSTRUCTION?
1116 004630  001407                     BEQ     2$
1117 004632  000137  041174             JMP     CPSPUR               ;IF NEITHER THEN REPORT
1118                                                                 ;UNEXPECTED TRAP TO 4.
1119
1120 004636  011637  001236    1$:      MOV     (SP),$TMP2
1121 004642  022626                     CMP     (SP)+,(SP)+
1122 004644  104014             15$:     ERROR   +14
1123 004646  000404                     BR      ADONE
1124
1125 004650  011637  001236    2$:      MOV     (SP),$TMP2
1126 004654  022626                     CMP     (SP)+,(SP)+
1127 004656  104015             25$:     ERROR   +15
1128
1129 004660                     ADONE:
     004660  104413                     RSETUP                       ;GO INITIALIZE THE FPS AND STACK; AND
                                                                     ;SEE IF THE USER HAS EXPRESSED
                                                                     ;THE DESIRE TO CHANGE THE SOFTWARE
                                                                     ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
                                                                     ;THE USER TYPED CONTROL G?).
1130
1131
```

```
                                                .SBTTL  TEST # 2 - CFCC TEST
                                                ;;*****************************************************************
                                                ;*TEST 2 - CFCC TEST
                                                ;*
                                                ;*THIS IS A TEST OF THE COPY CONDITION CODES INSTRUCTION, CFCC.
                                                ;*
                                                ;;*****************************************************************
     004662  000004                             TST2:   SCOPE
1138 004664  012737  004676  001110                     MOV     #B1,$LPERR      ;SET UP THE LOOP ON ERROR ADDRESS.
1139 004672  012700  000017                     MOV     #17,R0                  ;R0 CONTAINS TO TEST PATTERN.
1140
1141 004676                                     B1:
1142 004676  170100                             LDFPS   R0                      ;LOAD THE TEST PATTERN
1143
1144 004700                                     B2:
1145 004700  170000                             CFCC                            ;COPY CONDITION CODES.
1146
1147 004702  013703  177776                     MOV     PSW,R3                  ;SEE IF PATTERN TRANSFERED.
1148 004706  042703  177760                     BIC     #177760,R3
1149 004712  020003                             CMP     R0,R3
1150 004714  001002                             BNE     BERR
1151
1152 004716  077011                             B3:     SOB     R0,B1
1153 004720  000422                             BR      BDONE
1154
1155 004722                                     BERR:
1156 004722  170201                             STFPS   R1                      ;WAS FPS MODIFIED BY CFCC?
1157 004724  012737  004700  001236             MOV     #B2,$TMP2
1158 004732  020001                             CMP     R0,R1
1159 004734  001006                             BNE     BERR1
1160
1161 004736  010337  001240                     MOV     R3,$TMP3
1162 004742  010037  001242                     MOV     R0,$TMP4
1163 004746  104003                     1$:     ERROR   +3
1164 004750  000762                             BR      B3
1165
1166 004752                                     BERR1:
1167 004752  010037  001240                     MOV     R0,$TMP3
1168 004756  010137  001242                     MOV     R1,$TMP4
1169 004762  104004                     1$:     ERROR   +4
1170 004764  000754                             BR      B3
1171
1172 004766                                     BDONE:
     004766  104413                             RSETUP                          ;GO INITIALIZE THE FPS AND STACK; AND
                                                                                ;SEE IF THE USER HAS EXPRESSED
                                                                                ;THE DESIRE TO CHANGE THE SOFTWARE
                                                                                ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
                                                                                ;THE USER TYPED CONTROL G?).
1173
```

TEST # 3 - SETF, SETD, SETI AND SETL TEST

```
1182                                    .SBTTL  TEST # 3 - SETF, SETD, SETI AND SETL TEST
                                        ;;**********************************************************************
                                        ;*TEST 3 - SETF, SETD, SETI AND SETL TEST
                                        ;*
                                        ;*THIS IS A TEST OF THE SETF, SETD, SETI AND SETL INSTRUCTIONS.
                                        ;*EACH INSTRUCTION IS EXECUTED WITH THE FPS CONTAINING
                                        ;*ALL ONES AND ALSO WITH THE FPS CLEAR. THE RESULT OF EACH
                                        ;* SITUATION IS CHECKED.
                                        ;*
                                        ;;**********************************************************************
     004770  000004                     TST3:   SCOPE
1183 004772  012737  005006  001110             MOV     #C1,$LPERR      ;SET UP THE LOOP ON ERROR ADDRESS.
1184 005000  012737  000760  001244             MOV     #760,$TMP5
1185 005006  012737  000202  001250     C1:     MOV     #202,$TMP7
1186 005014  012737  042305  001252             MOV     #SETF1,$TMP10
1187 005022  005000                             CLR     R0
1188
1189 005024  170100                             LDFPS   R0                      ;CLEAR THE FPS.
1190 005026  012737  005034  001236             MOV     #C15,$TMP2
1191
1192 005034  170001                     C15:    SETF                            ;TEST INSTRUCTION.
1193
1194 005036  170201                             STFPS   R1                      ;GET RESULT.
1195 005040  005002                             CLR     R2
1196 005042  020201                             CMP     R2,R1                   ;DID AN ERROR OCCUR?
1197 005044  001402                             BEQ     1$
1198 005046  004737  005466                     JSR     PC,CERR1
1199
1200 005052                             1$:
     005052  012737  005060  001110             MOV     #C2,$LPERR      ;SET UP THE LOOP ON ERROR ADDRESS.
1201 005060  012700  147757             C2:     MOV     #147757,R0
1202
1203 005064  170100                             LDFPS   R0                      ;PUT 147757 IS FPS
1204 005066  012737  005074  001236             MOV     #C25,$TMP2
1205 005074  170001                     C25:    SETF                            ;CLEAR FD BIT.
1206
1207 005076  170201                             STFPS   R1                      ;GET RESULT
1208 005100  012702  147557                     MOV     #147557,R2
1209 005104  020102                             CMP     R1,R2                   ;RESULT CORRECT.
1210 005106  001402                             BEQ     1$
1211 005110  004737  005564                     JSR     PC,CERR2
1212
1213 005114                             1$:
     005114  012737  005122  001110             MOV     #C3,$LPERR      ;SET UP THE LOOP ON ERROR ADDRESS.
1214 005122  012737  000203  001250     C3:     MOV     #203,$TMP7
1215 005130  012737  042313  001252             MOV     #SETD1,$TMP10
1216 005136  012700  147757                     MOV     #147757,R0
1217
1218 005142  170100                             LDFPS   R0                      ;LOAD 147757 INTO FPS.
1219 005144  012737  005152  001236             MOV     #C35,$TMP2
1220 005152  170011                     C35:    SETD                            ;SETD FD BIT.
1221
1222 005154  170201                             STFPS   R1
1223 005156  012702  147757                     MOV     #147757,R2
1224 005162  020102                             CMP     R1,R2                   ;RESULT CORRECT?
1225 005164  001402                             BEQ     1$
1226 005166  004737  005564                     JSR     PC,CERR2
```

```
1227
1228 005172                          1$:
     005172  012737  005200  001110          MOV     #C4,$LPERR      ;SET UP THE LOOP ON ERROR ADDRESS.
1229 005200  005000                  C4:     CLR     R0
1230 005202  170100                          LDFPS   R0                      ;CLEAR FPS.
1231 005204  012737  005212  001236          MOV     #C45,$TMP2
1232
1233 005212  170011                  C45:    SETD                            ;SET FD BIT.
1234
1235 005214  170201                          STFPS   R1                      ;GET RESULT.
1236 005216  012702  000200                  MOV     #200,R2
1237 005222  020102                          CMP     R1,R2                   ;RESULT CORRECT?
1238 005224  001402                          BEQ     1$
1239 005226  004737  005466                  JSR     PC,CERR1
1240
1241 005232                          1$:
     005232  012737  005240  001110          MOV     #C5,$LPERR      ;SET UP THE LOOP ON ERROR ADDRESS.
1242 005240  012737  000204  001250  C5:     MOV     #204,$TMP7
1243 005246  012737  042321  001252          MOV     #SETI1,$TMP10
1244 005254  005000                          CLR     R0
1245
1246 005256  170100                          LDFPS   R0                      ;CLEAR FPS
1247 005260  012737  005266  001236          MOV     #C55,$TMP2
1248
1249 005266  170002                  C55:    SETI                            ;CLEAR FL BIT.
1250
1251 005270  170201                          STFPS   R1                      ;GET RESULT.
1252 005272  005002                          CLR     R2
1253 005274  020201                          CMP     R2,R1                   ;RESULT CORRECT?
1254 005276  001402                          BEQ     1$
1255 005300  004737  005466                  JSR     PC,CERR1
1256
1257 005304                          1$:
     005304  012737  005312  001110          MOV     #C6,$LPERR      ;SET UP THE LOOP ON ERROR ADDRESS.
1258 005312  012700  147757          C6:     MOV     #147757,R0
1259 005316  170100                          LDFPS   R0                      ;PUT 147757 INTO FPS
1260 005320  012737  005326  001236          MOV     #C65,$TMP2
1261
1262 005326  170002                  C65:    SETI                            ;CLEAR FL BIT.
1263
1264 005330  170201                          STFPS   R1                      ;GET THE RESULT.
1265 005332  012702  147657                  MOV     #147657,R2
1266 005336  020102                          CMP     R1,R2                   ;RESULT CORRECT?
1267 005340  001402                          BEQ     1$
1268 005342  004737  005564                  JSR     PC,CERR2
1269
1270 005346                          1$:
     005346  012737  005354  001110          MOV     #C7,$LPERR      ;SET UP THE LOOP ON ERROR ADDRESS.
1271 005354  012737  000205  001250  C7:     MOV     #205,$TMP7
1272 005362  012737  042327  001252          MOV     #SETL1,$TMP10
1273 005370  012700  147757                  MOV     #147757,R0
1274 005374  170100                          LDFPS   R0                      ;SET FPS TO 147757.
1275 005376  012737  005404  001236          MOV     #C75,$TMP2
1276
1277 005404  170012                  C75:    SETL                            ;SET FL BIT.
1278
1279 005406  170201                          STFPS   R1                      ;GET THE RESULT.
```

```
1280 005410  012702  147757                    MOV     #147757,R2
1281 005414  020102                            CMP     R1,R2                   ;RESULT CORRECT?
1282 005416  001402                            BEQ     1$
1283 005420  004737  005564                    JSR     PC,CERR2
1284
1285 005424                          1$:
     005424  012737  005432  001110            MOV     #C8,$LPERR      ;SET UP THE LOOP ON ERROR ADDRESS.
1286 005432  005000                  C8:       CLR     R0
1287 005434  170100                            LDFPS   R0                      ;CLEAR FPS.
1288 005436  012737  005444  001236            MOV     #C85,$TMP2
1289
1290 005444  170012                  C85:      SETL                            ;SET FL BIT.
1291
1292 005446  170201                            STFPS   R1
1293 005450  012702  000100                    MOV     #100,R2
1294 005454  020102                            CMP     R1,R2                   ;RESULT CORRECT.
1295 005456  001402                            BEQ     1$
1296 005460  004737  005466                    JSR     PC,CERR1
1297
1298 005464  000522                  1$:       BR      CDONE
1299
1300                                  ;THESE ARE ERROR ANALYSIS ROUTINES:
1301 005466  010103                  CERR1:    MOV     R1,R3
1302 005470  032703  177477                    BIT     #177477,R3              ;ARE ANY OTHER BITS SET?
1303 005474  001401                            BEQ     2$
1304 005476  000503                  1$:       BR      CERR4
1305
1306 005500  022703  000300          2$:       CMP     #300,R3                 ;ARE ROTH FD AND FL SET?
1307 005504  001774                            BEQ     1$
1308 005506  032703  000300                    BIT     #300,R3                 ;ARE THEY BOTH CLEAR?
1309 005512  001771                            BEQ     1$
1310
1311 005514  032703  000200                    BIT     #200,R3                 ;IS FD SET?
1312 005520  001407                            BEQ     3$
1313 005522  012737  042313  001254            MOV     #SETD1,$TMP11
1314 005530  012737  000203  001246            MOV     #203,$TMP6
1315 005536  000452                            BR      CERR3
1316
1317 005540  032703  000100          3$:       BIT     #100,R3                 ;IS FL SET
1318 005544  001754                            BEQ     1$
1319 005546  012737  042327  001254            MOV     #SETL1,$TMP11
1320 005554  012737  000205  001246            MOV     #205,$TMP6
1321 005562  000440                            BR      CERR3
1322
1323 005564  010103                  CERR2:    MOV     R1,R3
1324 005566  005103                            COM     R3
1325
1326 005570  032703  177477                    BIT     #177477,R3              ;ARE ANY OTHER BITS SET?
1327 005574  001401                            BEQ     2$
1328 005576  000443                  1$:       BR      CERR4
1329
1330 005600  032703  000300          2$:       BIT     #300,R3                 ;ARE BOTH FD AND FL SET?
1331 005604  001774                            BEQ     1$
1332 005606  032701  000300                    BIT     #300,R1                 ;ARE THEY BOTH CLEAR?
1333 005612  001771                            BEQ     1$
1334
1335 005614  032701  000200                    BIT     #200,R1                 ;IS FD CLEAR?
```

```
1336 005620  001007                        BNE     3$
1337 005622  012737  042305  001254        MOV     #SETF1,$TMP11
1338 005630  012737  000202  001246        MOV     #202,$TMP6
1339 005636  000412                        BR      CERR3
1340
1341 005640  032701  000100          3$:   BIT     #100,R1
1342 005644  001354                        BNE     1$                      ;IS FL CLEAR.
1343 005646  012737  042321  001254        MOV     #SETI1,$TMP11
1344 005654  012737  000204  001246        MOV     #204,$TMP6
1345 005662  000400                        BR      CERR3
1346
1347                              ;REPORT THE ERRORS:
1348 005664                       CERR3:
1349 005664  010137  001240                MOV     R1,$TMP3
1350 005670  010237  001242                MOV     R2,$TMP4
1351 005674  012637  005730                MOV     (SP)+,CPC
1352 005700  104012               1$:     ERROR   +12
1353 005702  000177  000022                JMP     @CPC
1354
1355 005706                       CERR4:
1356 005706  010137  001240                MOV     R1,$TMP3
1357 005712  010237  001242                MOV     R2,$TMP4
1358 005716  012637  005730                MOV     (SP)+,CPC
1359 005722  104013               1$:     ERROR   +13
1360 005724  000177  000000                JMP     @CPC
1361
1362 005730  000000               CPC:    .WORD   0
1363
1364 005732                       CDONE:
     005732  104413                        RSETUP                  ;GO INITIALIZE THE FPS AND STACK; AND
                                                                    ;SEE IF THE USER HAS EXPRESSED
                                                                    ;THE DESIRE TO CHANGE THE SOFTWARE
                                                                    ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
                                                                    ;THE USER TYPED CONTROL G?).
1365
1366
```

```
1386                                    .SBTTL  TEST # 4 - ILLEGAL FPP OP CODES AND STST TEST
                                        ;;*********************************************************************
                                        ;*TEST 4 - ILLEGAL FPP OP CODES AND STST TEST
                                        ;*
                                        ;*THIS IS A TEST OF THE FPP OPERATION CODES:
                                        ;*                      170003
                                        ;*                      170004
                                        ;*                        :
                                        ;*                      170010
                                        ;*                      170013
                                        ;*                      170014
                                        ;*                        :
                                        ;*                      170077
                                        ;*THESE ARE ILLEGAL INSTRUCTIONS AND (WITH INTERRUPTS ENABLED)
                                        ;*SHOULD CAUSE A TRAP TO 244.
                                        ;*ALSO TESTED HERE IS THE INSTRUCTION:
                                        ;*              STST    R1
                                        ;*WHICH SHOULD PUT THE FEC CODE 2 IN R1, AFTER ANY OF THE ABOVE
                                        ;*OP CODES IS EXECUTED.
                                        ;*
                                        ;*********************************************************************
     005734  000004                     TST4:   SCOPE
1387 005736  012737  005764  001110             MOV     #D1,$LPERR      ;SET UP THE LOOP ON ERROR ADDRESS.
1388 005744  012705  170003                     MOV     #170003,R5              ;INITIAL OP CODE.
1389 005750  012737  006154  000004             MOV     #DERR2,ERRVECT
1390 005756  012737  006060  000244             MOV     #DERR1,FPVECT
1391
1392 005764  005000                     D1:     CLR     R0
1393 005766  170100                             LDFPS   R0                      ;CLEAR FPS.
1394 005770  005002                             CLR     R2
1395 005772  010537  006010                     MOV     R5,D2                   ;SET UP THE ILLEGAL INSTRUCTION.
1396 005776  010537  001244                     MOV     R5,$TMP5
1397 006002  012737  006010  001236             MOV     #D2,$TMP2
1398 006010  000000                     D2:     .WORD   0
1399 006012  170000                     D3:     CFCC
1400 006014  005202                             INC     R2
1401 006016  005202                     D4:     INC     R2
1402
1403 006020  170201                             STFPS   R1                      ;REPORT FAILURE. DID NOT TRAP.
1404 006022  010137  001240                     MOV     R1,$TMP3
1405 006026  104016                     1$:     ERROR   +16
1406
1407 006030  022705  170010             D5:     CMP     #170010,R5              ;COMPUTE NEXT OP CODE
1408 006034  001003                             BNE     D6
1409 006036  012705  170013                     MOV     #170013,R5
1410 006042  000750                             BR      D1
1411
1412 006044  022705  170077             D6:     CMP     #170077,R5
1413 006050  001001                             BNE     D7
1414 006052  000452                             BR      DDONE
1415 006054  005205                     D7:     INC     R5
1416 006056  000742                             BR      D1
1417
1418 006060  022716  006012             DERR1:  CMP     #D3,(SP)                ;DID TRAP OCCUR ON TEST INSTRUCTION?
1419 006064  001402                             BEQ     1$
1420 006066  000137  041142                     JMP     FPSPUR
1421
```

```
1422 006072  022626                  1$:    CMP     (SP)+,(SP)+
1423 006074  170201                         STFPS   R1                  ;GET THE FPS AND SEE IF IT IS
1424 006076  022701  100000                 CMP     #100000,R1          ;SET CORRECTLY.
1425 006102  001406                         BEQ     3$
1426
1427 006104  012737  100000  001240         MOV     #100000,$TMP3
1428 006112  010137  001242                 MOV     R1,$TMP4
1429 006116  104017                  2$:    ERROR   +17
1430
1431 006120  012704  000001          3$:    MOV     #1,R4
1432 006124  170304                  D8:    STST    R4                  ;GET THE FEC CODE. NOTE THAT
1433                                                                     ;IF THE DESTINATION MODE IS
1434                                                                     ;IMPROPERLY DECODED AN ODD
1435                                                                     ;ADDRESS TRAP TO 4 SHOULD OCCUR.
1436 006126  022704  000002                 CMP     #2,R4               ;WAS FEC CORRECT?
1437 006132  001001                         BNE     D9
1438 006134  000735                         BR      D5
1439
1440 006136                          D9:                                ;REPORT STST FAILURE
1441 006136  012737  006124  001240         MOV     #D8,$TMP3
1442 006144  010437  001242                 MOV     R4,$TMP4
1443 006150  104020                  1$:    ERROR   +20
1444 006152  000726                         BR      D5
1445
1446 006154  022716  006126          DERR2: CMP     #D8+2,(SP)          ;DID THE TRAP OCCUR ON THE
1447 006160  001402                         BEQ     D10                 ;STST INSTRUCTION?
1448 006162  000137  041174                 JMP     CPSPUR
1449
1450 006166                          D10:
1451 006166  011637  001236                 MOV     (SP),$TMP2
1452 006172  022626                         CMP     (SP)+,(SP)+
1453 006174  104021                  1$:    ERROR   +21
1454 006176  000714                         BR      D5
1455
1456 006200                          DDONE:
     006200  104413                         RSETUP                      ;GO INITIALIZE THE FPS AND STACK; AND
                                                                         ;SEE IF THE USER HAS EXPRESSED
                                                                         ;THE DESIRE TO CHANGE THE SOFTWARE
                                                                         ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
                                                                         ;THE USER TYPED CONTROL G?).
1457
1458
```

```
                                        .SBTTL  TEST # 5 - FID, INTERRUPT DISABLE, BIT TEST
                                        ;;*****************************************************************
                                        ;*TEST 5 - FID, INTERRUPT DISABLE, BIT TEST
                                        ;*
                                        ;*THIS IS A TEST OF FPS BIT 14 (FID) OR FLOATING INTERRUPT DISABLE.
                                        ;*AN ILLEGAL INSTRUCTION IS EXECUTED WITH FID=1. NO INTERRUPT SHOULD
                                        ;*OCCUR.
                                        ;*
                                        ;;*****************************************************************
006202  000004                          TST5:   SCOPE
1467 006204  012737  006220  001110             MOV     #E1,$LPERR      ;SET UP THE LOOP ON ERROR ADDRESS.
1468 006212  012737  006314  000244             MOV     #EERR2,FPVECT           ;SETUP FOR THE INTERRUPT.
1469
1470 006220  012700  040000          E1:        MOV     #40000,R0
1471 006224  170100                             LDFPS   R0                      ;SET FID.
1472 006226  012737  006234  001236             MOV     #E3,$TMP2
1473 006234                          E2:
1474 006234  170020                  E3:        .WORD   170020                  ;ILLEGAL FPP INSTRUCTION.
1475 006236  170000                  E4:        CFCC
1476
1477 006240  170201                             STFPS   R1                      ;SEE IF ERROR WAS DETECTED.
1478 006242  022701  140000                     CMP     #140000,R1
1479 006246  001005                             BNE     EERR0
1480
1481 006250  170304                             STST    R4                      ;SEE IF FEC=2
1482 006252  022704  000002                     CMP     #2,R4
1483 006256  001010                             BNE     EERR1
1484 006260  000431                             BR      EDONE
1485
1486 006262                          EERR0:                                     ;REPORT FPS INCORRECTLY SET.
1487 006262  010137  001240                     MOV     R1,$TMP3
1488 006266  012737  140000  001242             MOV     #140000,$TMP4
1489 006274  104022                  1$:        ERROR   +22
1490 006276  000422                             BR      EDONE
1491
1492 006300                          EERR1:                                     ;REPORT FEC NOT 2.
1493 006300  010537  001240                     MOV     R5,$TMP3
1494 006304  010437  001242                     MOV     R4,$TMP4
1495 006310  104023                  1$:        ERROR   +23
1496 006312  000414                             BR      EDONE
1497
1498 006314  021627  006236          EERR2:     CMP     (SP),#E4                ;DID THE ILLEGAL INSTRUCTION TRAP?
1499 006320  001402                             BEQ     1$
1500 006322  000137  041142                     JMP     FPSPUR
1501
1502 006326                          1$:
1503 006326  011637  001236                     MOV     (SP),$TMP2
1504 006332  022626                             CMP     (SP)+,(SP)+
1505 006334  170201                             STFPS   R1
1506 006336  010137  001240                     MOV     R1,$TMP3
1507 006342  104024                  2$:        ERROR   +24
1508
1509 006344                          EDONE:
     006344  104413                             RSETUP                  ;GO INITIALIZE THE FPS AND STACK; AND
                                                                         ;SEE IF THE USER HAS EXPRESSED
                                                                         ;THE DESIRE TO CHANGE THE SOFTWARE
                                                                         ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
```

```
                                                        ;THE USER TYPED CONTROL G?).

1510
1511
```

```
1524                                    .SBTTL  TEST # 6 - LDD AND STD, WITH SRC AND DST MODE 1, TEST
                                        ;;**********************************************************************
                                        ;*TEST 6 - LDD AND STD, WITH SRC AND DST MODE 1, TEST
                                        ;*
                                        ;*THIS IS A TEST OF BOTH THE INSTRUCTION:
                                        ;*              LDD     (R0),AC0
                                        ;*AND THE INSTRUCTION:
                                        ;*              STD     AC0,(R0)
                                        ;*MOST OF THE FAILURES ARE ISOLATED TO THE SRC OR DST FLOWS. NOTE
                                        ;*THAT THE INTEGRITY OF AC0 HAS NOT BEEN ASSURED. THIS MEANS THAT
                                        ;*IN SOME CASES IT WILL BE IMPOSSIBLE TO ISOLATE CERTAIN DATA PATTERN
                                        ;*FAILURES TO EITHER THE FLOWS OR THIS ACCUMULATOR.
                                        ;*
                                        ;;**********************************************************************
    006346  000004                      TST6:   SCOPE
1525
1526 006350                             F1:
    006350  012737  006350  001110              MOV     #F1,$LPERR      ;SET UP THE LOOP ON ERROR ADDRESS.
1527 006356  012737  006426  001236              MOV     #F3,$TMP2
1528 006364  005000                             CLR     R0
1529 006366  170100                             LDFPS   R0
1530 006370  170011                             SETD
1531 006372  012701  010164                     MOV     #FDATI0,R1              ;SET UP THE LOAD DATA.
1532 006376  012702  010230                     MOV     #FXDAT0,R2
1533 006402  012703  000010                     MOV     #10,R3
1534
1535 006406  012221                     F2:     MOV     (R2)+,(R1)+
1536 006410  077302                             SOB     R3,F2
1537
1538 006412  012700  010174                     MOV     #FDATI4,R0              ;SETUP R0 FOR THE LDD (R0),AC0.
1539 006416  012737  007650  000004              MOV     #FERR20,ERRVECT         ;IF THE SRC FLOWS FAIL THEN
1540                                                                             ;AN ODD ADDRESS MAY OCCUR.
1541 006424  005003                             CLR     R3
1542
1543 006426  172410                     F3:     LDD     (R0),AC0
1544 006430  005203                     F4:     INC     R3
1545 006432  005203                             INC     R3
1546
1547 006434  020027  010174                     CMP     R0,#FDATI4              ;WAS R0 AFFECTED?
1548 006440  001402                             BEQ     F5
1549 006442  000137  007014                     JMP     FERR1
1550
1551 006446  020327  000002             F5:     CMP     R3,#2                   ;SEE IF THE PC WAS ADVERSELY
1552 006452  001402                             BEQ     1$                      ;AFFECTED DURING THE INSTRUCTION.
1553 006454  000137  007112                     JMP     FERR2
1554
1555 006460  012701  010164             1$:     MOV     #FDATI0,R1              ;MAKE SURE THE SOURCE DATA WAS
1556 006464  012702  010230                     MOV     #FXDAT0,R2              ;NOT AFFECTED.
1557 006470  012703  000010                     MOV     #10,R3
1558 006474  022122                     2$:     CMP     (R1)+,(R2)+
1559 006476  001402                             BEQ     3$
1560 006500  000137  006756                     JMP     FERR0
1561 006504  077305                     3$:     SOB     R3,2$
1562
1563 006506  170201                             STFPS   R1                      ;MAKE SURE THE FPS IS CORRECT.
1564 006510  022701  000200                     CMP     #200,R1
1565 006514  001402                             BEQ     F6
```

```
1566 006516  000137  007630                    JMP     FERR11
1567
1568 006522                          F6:
     006522  012737  006530  001110             MOV     #1$,$LPERR      ;SET UP THE LOOP ON ERROR ADDRESS.
1569 006530  012737  006572  001236  1$:        MOV     #F10,$TMP2
1570
1571 006536  012703  177777                      MOV     #-1,R3
1572 006542  012704  000010                      MOV     #10,R4
1573 006546  012705  010206                      MOV     #FDAT00,R5              ;SET UP THE OUTPUT DATA BUFFER.
1574 006552  010325                    F7:       MOV     R3,(R5)+
1575 006554  077402                              SOB     R4,F7
1576
1577 006556  012700  010216                      MOV     #FDAT04,R0              ;SET UP R0 FOR DST MODE 1 REG 0.
1578 006562  012737  010016  000004              MOV     #FERR25,ERRVECT         ;IF THE DST FLOWS FAIL AN ODD
1579                                                                            ;ADDRESS COULD OCCUR.
1580 006570  005003                              CLR     R3
1581
1582 006572  174010                    F10:      STD     AC0,(R0)                ;TEST INSTRUCTION.
1583 006574  005203                    F11:      INC     R3
1584 006576  005203                              INC     R3
1585
1586 006600  020027  010216                      CMP     R0,#FDAT04              ;WAS R0 MODIFIED?
1587 006604  001402                              BEQ     F12
1588 006606  000137  007152                      JMP     FERR3
1589
1590 006612  020327  000002                    F12:      CMP     R3,#2                   ;WAS THE PC AFFECTED CORRECTLY?
1591 006616  001402                              BEQ     F135
1592 006620  000137  007144                      JMP     FERR4
1593
1594 006624  012701  010206                    F135:     MOV     #FDAT00,R1
1595 006630  012702  010230                      MOV     #FXDAT0,R2
1596
1597 006634  022122                              CMP     (R1)+,(R2)+             ;SEE IF THE DATA WAS OUTPUT
1598 006636  001402                              BEQ     F13                     ;TO THE TARGET AREA CORRECTLY.
1599 006640  000137  007250                      JMP     FERR5
1600
1601 006644  022122                    F13:      CMP     (R1)+,(R2)+
1602 006646  001402                              BEQ     F14
1603 006650  000137  007250                      JMP     FERR5
1604
1605 006654  022122                    F14:      CMP     (R1)+,(R2)+
1606 006656  001402                              BEQ     F15
1607 006660  000137  007250                      JMP     FERR5
1608
1609 006664  022122                    F15:      CMP     (R1)+,(R2)+
1610 006666  001402                              BEQ     F16
1611 006670  000137  007250                      JMP     FERR5
1612
1613 006674  022122                    F16:      CMP     (R1)+,(R2)+
1614 006676  001402                              BEQ     F17
1615 006700  000137  007574                      JMP     FERR10
1616
1617 006704  022122                    F17:      CMP     (R1)+,(R2)+
1618 006706  001402                              BEQ     F20
1619 006710  000137  007304                      JMP     FERR6
1620
1621 006714  022122                    F20:      CMP     (R1)+,(R2)+
```

```
1622 006716  001402                        BEQ     F21
1623 006720  000137  007440                JMP     FERR7
1624
1625 006724  022122                F21:    CMP     (R1)+,(R2)+
1626 006726  001402                        BEQ     F22
1627 006730  000137  007574                JMP     FERR10
1628
1629 006734  005001                F22:    CLR     R1
1630 006736  170201                        STFPS   R1                      ;MAKE SURE FPS IS CORRECT.
1631 006740  022701  000200                CMP     #200,R1
1632 006744  001402                        BEQ     F23
1633 006746  000137  007630                JMP     FERR11
1634 006752  000137  010250          F23:  JMP     FDONE
1635
1636 006756                        FERR0:                                  ;SOURCE DATA AFFECTED BY
1637 006756  012737  010230  001240        MOV     #FXDAT0,$TMP3           ;THE LDD INSTRUCTION.
1638 006764  012737  010242  001242        MOV     #FXDAT0+12,$TMP4
1639 006772  012737  010164  001244        MOV     #FDATI0,$TMP5
1640 007000  012737  010176  001246        MOV     #FDATI0+12,$TMP6
1641 007006  104025                1$:     ERROR   +25
1642 007010  000137  010250                JMP     FDONE
1643
1644 007014  012737  010174  001242  FERR1: MOV     #FDATI4,$TMP4           ;FSRC FLOWS FAILURE.
1645 007022  010037  001240                MOV     R0,$TMP3
1646 007026  012737  000762  001244        MOV     #762,$TMP5
1647 007034  012737  000321  001250        MOV     #321,$TMP7
1648
1649 007042  022700  010164                CMP     #FDATI0,R0              ;FSRC MODE 4?
1650 007046  001004                        BNE     1$
1651 007050  012737  000324  001246        MOV     #324,$TMP6
1652 007056  000412                        BR      4$
1653
1654 007060  022700  010204          1$:   CMP     #FDATI4+10,R0           ;FSRC MODE 2?
1655 007064  001004                        BNE     2$
1656 007066  012737  000322  001246        MOV     #322,$TMP6
1657 007074  000403                        BR      4$
1658
1659 007076                          2$:
1660 007076  104027                3$:     ERROR   +27
1661 007100  000137  010250                JMP     FDONE
1662
1663 007104                          4$:
1664 007104  104026                5$:     ERROR   +26
1665 007106  000137  010250                JMP     FDONE
1666
1667 007112  012701  006430          FERR2: MOV     #F4,R1                  ;THE PC WAS INCORRECTLY AFFECTED
1668                                                                        ;DURING THE INSTRUCTION.
1669 007116  010137  001242          FER2:  MOV     R1,$TMP4
1670 007122  162701  000004                SUB     #4,R1
1671 007126  006303                        ASL     R3
1672 007130  060301                        ADD     R3,R1
1673 007132  010137  001240                MOV     R1,$TMP3
1674 007136  104030                1$:     ERROR   +30
1675 007140  000137  010250                JMP     FDONE
1676
1677 007144  012701  006574          FERR4: MOV     #F11,R1
1678 007150  000762                        BR      FER2
```

```
1679
1680 007152  012737  010216  001242  FERR3:  MOV     #FDAT04,$TMP4           ;FAILURE IN THE FDST FLOWS.
1681 007160  010037  001240                  MOV     R0,$TMP3
1682 007164  012737  000527  001244          MOV     #527,$TMP5
1683 007172  012737  000641  001250          MOV     #641,$TMP7
1684
1685 007200  022700  010206                  CMP     #FDAT00,R0              ;DST MODE 4?
1686 007204  001004                          BNE     1$
1687 007206  012737  000644  001246          MOV     #644,$TMP6
1688 007214  000412                          BR      4$
1689
1690 007216  022700  010226          1$:     CMP     #FDAT04+10,R0           ;DST MODE 2?
1691 007222  001004                          BNE     2$
1692 007224  012737  000642  001246          MOV     #642,$TMP6
1693 007232  000403                          BR      4$
1694
1695 007234                          2$:
1696 007234  104032                  3$:     ERROR   +32
1697 007236  000137  010250                  JMP     FDONE
1698
1699 007242                          4$:
1700 007242  104031                  5$:     ERROR   +31
1701 007244  000137  010250                  JMP     FDONE
1702
1703 007250                          FERR5:                                  ;FAILURE OF STD.
1704 007250  010037  001240                  MOV     R0,$TMP3
1705 007254  012737  010206  001242          MOV     #FDAT00,$TMP4
1706 007262  012737  010224  001244          MOV     #FDAT07,$TMP5
1707 007270  012737  010230  001246          MOV     #FXDAT0,$TMP6
1708 007276  104033                  1$:     ERROR   +33
1709 007300  000137  010250                  JMP     FDONE
1710
1711 007304  012701  010220          FERR6:  MOV     #FDAT05,R1              ;DID (BUT GR7) FAIL IN THE FDST
1712 007310  012702  177777                  MOV     #-1,R2                  ;FLOWS?
1713 007314  012703  000003                  MOV     #3,R3
1714 007320  020221                  1$:     CMP     R2,(R1)+
1715 007322  001017                          BNE     5$
1716 007324  077303                          SOB     R3,1$
1717
1718                                                                         ;REPORT FAILURE OF (BUT GR7) IN
1719 007326  010037  001240                  MOV     R0,$TMP3                ;THE FDST FLOWS.
1720 007332  012737  000412  001244          MOV     #412,$TMP5
1721 007340  012737  000147  001246          MOV     #147,$TMP6
1722 007346  012737  000145  001250          MOV     #145,$TMP7
1723 007354  104034                  2$:     ERROR   +34
1724 007356  000137  010250                  JMP     FDONE
1725
1726 007362  012701  010220          5$:     MOV     #FDAT05,R1              ;DID (BUT GR7) FAIL IN THE SRC FLOWS?
1727 007366  012703  000003                  MOV     #3,R3
1728 007372  005721                  6$:     TST     (R1)+
1729 007374  001402                          BEQ     7$
1730 007376  000137  007574                  JMP     FERR10
1731 007402  077305                  7$:     SOB     R3,6$
1732
1733                                                                         ;REPORT FAILURE OF (BUT GR7) IN
1734 007404  010037  001240                  MOV     R0,$TMP3                ;THE FSRC FLOWS.
1735 007410  012737  000207  001244          MOV     #207,$TMP5
```

```
1736 007416  012737  000176  001246          MOV     #176,$TMP6
1737 007424  012737  000174  001250          MOV     #174,$TMP7
1738
1739 007432  104035                  10$:    ERROR   +35
1740 007434  000137  010250                  JMP     FDONE
1741
1742 007440  012701  010222          FERR7:  MOV     #FDAT06,R1          ;DID (BUT FD) FAIL IN THE FDST FLOWS?
1743 007444  012702  177777                  MOV     #-1,R2
1744 007450  012703  000002                  MOV     #2,R3
1745 007454  020221                  1$:     CMP     R2,(R1)+
1746 007456  001017                          BNE     5$
1747 007460  077303                          SOB     R3,1$
1748
1749                                                                     ;REPORT FAILURE OF (BUT FD) IN THE
1750 007462  010037  001240                  MOV     R0,$TMP3            ;FDST FLOWS.
1751 007466  012737  000707  001244          MOV     #707,$TMP5
1752 007474  012737  000244  001246          MOV     #244,$TMP6
1753 007502  012737  000245  001250          MOV     #245,$TMP7
1754 007510  104036                  2$:     ERROR   +36
1755 007512  000137  010250                  JMP     FDONE
1756
1757 007516  012701  010222          5$:     MOV     #FDAT06,R1          ;DID (BUT FD) FAIL IN THE FSRC FLOWS?
1758 007522  012703  000002                  MOV     #2,R3
1759 007526  005721                  6$:     TST     (R1)+
1760 007530  001402                          BEQ     7$
1761 007532  000137  007574                  JMP     FERR10
1762 007536  077305                  7$:     SOB     R3,6$
1763
1764                                                                     ;REPORT FAILURE OF (BUT FD) IN THE
1765 007540  010037  001240                  MOV     R0,$TMP3            ;FSRC FLOWS.
1766 007544  012737  000441  001244          MOV     #441,$TMP5
1767 007552  012737  000076  001246          MOV     #76,$TMP6
1768 007560  012737  000077  001250          MOV     #77,$TMP7
1769 007566  104037                  10$:    ERROR   +37
1770 007570  000137  010250                  JMP     FDONE
1771
1772 007574                          FERR10:                             ;REPORT DATA ERROR.
1773 007574  010037  001240                  MOV     R0,$TMP3
1774 007600  012737  010216  001242          MOV     #FDAT04,$TMP4
1775 007606  012737  010224  001244          MOV     #FDAT07,$TMP5
1776 007614  012737  010240  001246          MOV     #FXDAT4,$TMP6
1777 007622  104040                  1$:     ERROR   +40
1778 007624  000137  010250                  JMP     FDONE
1779
1780 007630                          FERR11:                             ;REPORT BAD FPS.
1781 007630  010137  001240                  MOV     R1,$TMP3
1782 007634  012737  000200  001242          MOV     #200,$TMP4
1783 007642  104041                  1$:     ERROR   +41
1784 007644  000137  010250                  JMP     FDONE
1785
1786 007650  012737  041345  001264  FERR20: MOV     #NULL,$TMP15        ;THE EXECUTION OF THE LDD
1787 007656  005037  001252                  CLR     $TMP10              ;CAUSED A TRAP TO 4, BECAUSE
1788 007662  011637  001236                  MOV     (SP),$TMP2          ;A FSRC FLOW FAILURE RESULTED
1789 007666  012737  010174  001240          MOV     #FDATI4,$TMP3       ;IN AN ODD ADDRESS.
1790 007674  012737  000321  001250          MOV     #321,$TMP7
1791 007702  012737  000762  001244          MOV     #762,$TMP5
1792
```

```
1793 007710  021627  006432                    CMP     (SP),#F4+2              ;SEE IF FSRC MODE 6 OR 7 WAS
1794 007714  001424                            BEQ     FERR21                  ;EXECUTED.
1795
1796 007716  020027  010172                    CMP     R0,#FDAT13              ;FSRC MODE 5?
1797 007722  001006                            BNE     2$
1798
1799                                                                           ;REPORT FSRC FLOW FAILURE TO
1800 007724  012737  000325  001246            MOV     #325,$TMP6              ;MODE 5.
1801 007732  022626                            CMP     (SP)+,(SP)+
1802 007734  104042                    1$:     ERROR   +42
1803 007736  000544                            BR      FDONE
1804
1805 007740  020027  010176            2$:     CMP     R0,#FDAT15              ;FSRC MODE 3?
1806 007744  001402                            BEQ     3$
1807 007746  000137  041174                    JMP     CPSPUR
1808
1809 007752                            3$:                                     ;REPORT FSRC     FLOW FAILURE TO
1810 007752  012737  000323  001246            MOV     #323,$TMP6              ;MODE 3.
1811 007760  022626                            CMP     (SP)+,(SP)+
1812 007762  104042                    4$:     ERROR   +42
1813 007764  000531                            BR      FDONE
1814
1815 007766  022626                    FERR21: CMP     (SP)+,(SP)+             ;REPORT FSRC FLOW FAILURE TO
1816                                                                           ;MODE 6 OR MODE 7.
1817 007770  012737  043111  001264            MOV     #MS16,$TMP15
1818 007776  012737  000326  001246            MOV     #326,$TMP6
1819 010004  012737  000327  001252            MOV     #327,$TMP10
1820 010012  104042                    1$:     ERROR   +42
1821 010014  000515                            BR      FDONE
1822
1823 010016  012737  041345  001264    FERR25: MOV     #NULL,$TMP15            ;THE EXECUTION OF THE STD INSTRUCTION
1824 010024  005037  001252                    CLR     $TMP10                  ;TRAPPED TO 4, BECAUSE A FAILURE
1825 010030  012737  010216  001240            MOV     #FDAT04,$TMP3           ;IN THE FDST FLOWS RESULTED
1826 010036  011637  001236                    MOV     (SP),$TMP2              ;IN AN ODD ADDRESS.
1827 010042  012737  000527  001244            MOV     #527,$TMP5
1828 010050  012737  000641  001250            MOV     #641,$TMP7
1829
1830 010056  021627  006574                    CMP     (SP),#F10+2             ;FLOW FAILURE TO FDST MODE 6 OR 7?
1831 010062  001424                            BEQ     FERR26
1832
1833 010064  020027  010214                    CMP     R0,#FDAT03              ;DID FDST FLOW FAIL TO MODE 5?
1834 010070  001006                            BNE     2$
1835
1836                                                                           ;REPORT FLOW FAILURE TO FDST
1837 010072  012737  000645  001246            MOV     #645,$TMP6              ;MODE 5.
1838 010100  022626                            CMP     (SP)+,(SP)+
1839 010102  104043                    1$:     ERROR   +43
1840 010104  000461                            BR      FDONE
1841
1842 010106  020027  010220            2$:     CMP     R0,#FDAT05              ;DID FDST FLOW FAIL TO MODE 3?
1843 010112  001402                            BEQ     3$
1844 010114  000137  041174                    JMP     CPSPUR
1845
1846 010120                            3$:                                     ;REPORT FDST FLOW FAILED TO MODE 3.
1847 010120  012737  000643  001246            MOV     #643,$TMP6
1848 010126  022626                            CMP     (SP)+,(SP)+
1849 010130  104043                    4$:     ERROR   +43
```

```
1850 010132  000446                          BR      FDONE
1851
1852 010134                          FERR26:                         ;REPORT FDST FLOW FAILURE TO MODE
1853 010134  012737  043111  001264          MOV     #MS16,$TMP15    ;6 OR MODE 7.
1854 010142  012737  000646  001246          MOV     #646,$TMP6
1855 010150  012737  000647  001252          MOV     #647,$TMP10
1856 010156  022626                          CMP     (SP)+,(SP)+
1857 010160  104043                  1$:     ERROR   +43
1858 010162  000432                          BR      FDONE
1859
1860 010164  177777                  FDATI0: -1
1861 010166  177777                  FDATI1: -1
1862 010170  177777                  FDATI2: -1
1863 010172  177777                  FDATI3: -1
1864 010174  177777                  FDATI4: -1
1865 010176  177777                  FDATI5: -1
1866 010200  177777                  FDATI6: -1
1867 010202  177777                  FDATI7: -1
1868 010204  177777                          -1
1869 010206  177777                  FDATO0: -1
1870 010210  177777                  FDATO1: -1
1871 010212  177777                  FDATO2: -1
1872 010214  177777                  FDATO3: -1
1873 010216  177777                  FDATO4: -1
1874 010220  177777                  FDATO5: -1
1875 010222  177777                  FDATO6: -1
1876 010224  177777                  FDATO7: -1
1877 010226  177777                          -1
1878 010230  177777                  FXDAT0: -1
1879 010232  177777                  FXDAT1: -1
1880 010234  177777                  FXDAT2: -1
1881 010236  177777                  FXDAT3: -1
1882 010240  052525                  FXDAT4: 052525
1883 010242  031463                  FXDAT5: 031463
1884 010244  007417                  FXDAT6: 007417
1885 010246  000477                  FXDAT7: 000477
1886
1887
1888 010250                           FDONE:
     010250  104413                          RSETUP                  ;GO INITIALIZE THE FPS AND STACK; AND
                                                                     ;SEE IF THE USER HAS EXPRESSED
                                                                     ;THE DESIRE TO CHANGE THE SOFTWARE
                                                                     ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
                                                                     ;THE USER TYPED CONTROL G?).
1889
1890
```

```
                                .SBTTL  TEST # 7 - FSRC MODE 0 TEST
                                ;;*****************************************************************
                                ;*TEST 7 - FSRC MODE 0 TEST
                                ;*
                                ;*THIS IS A TEST OF FSRC MODE ZERO USING THE LDD AND LDF INSTRUCTIONS.
                                ;*
                                ;;*****************************************************************
 010252  000004                 TST7:   SCOPE
1897 010254  012737  010262  001110        MOV     #I1,$LPERR      ;SET UP THE LOOP ON ERROR ADDRESS.
1898
1899 010262                     I1:
1900 010262  170011                        SETD                    ;SET FD.
1901 010264  012700  011112               MOV     #IDAT10,R0
1902 010270  012701  011062               MOV     #IPAT10,R1
1903 010274  012702  000004               MOV     #4,R2
1904 010300  012120             I2:       MOV     (R1)+,(R0)+     ;SET UP THE INPUT DATA BUFFER.
1905 010302  077202                       SOB     R2,I2
1906
1907 010304  012700  011112               MOV     #IDAT10,R0      ;LOAD AC1
1908 010310  172510                       LDD     (R0),AC1
1909
1910 010312  012700  011072               MOV     #IPAT20,R0      ;LOAD AC0
1911 010316  172410                       LDD     (R0),AC0
1912
1913 010320  012701  000001               MOV     #1,R1           ;IN CASE THE FSRC FLOWS FAIL
1914 010324  012737  010662  000004       MOV     #IERR0,ERRVECT  ;AN ODD ADDRESS TRAP TO 4 MAY OCCUR.
1915 010332  012737  010346  001236       MOV     #I3,$TMP2
1916 010340  012737  043571  001240       MOV     #MS35,$TMP3
1917 010346  172401             I3:       LDD     AC1,AC0         ;TEST INSTRUCTION.
1918 010350  000240             I4:       NOP
1919 010352  000240             I5:       NOP
1920
1921 010354  012700  011102               MOV     #IDAT00,R0
1922 010360  174010                       STD     AC0,(R0)        ;GET AC0, THE RESULTS.
1923
1924 010362  012700  011102               MOV     #IDAT00,R0      ;SEE IF DATA IS CORRECT.
1925 010366  012701  011112               MOV     #IDAT10,R1
1926 010372  012702  000004               MOV     #4,R2
1927 010376  022021             I6:       CMP     (R0)+,(R1)+
1928 010400  001424                       BEQ     I105
1929
1930 010402  012700  011106               MOV     #IDAT02,R0      ;SEE IF (BUT FD) FAILED.
1931 010406  012702  000002               MOV     #2,R2
1932 010412  005720             I7:       TST     (R0)+
1933 010414  001413                       BEQ     I10
1934
1935 010416  012700  011106               MOV     #IDAT02,R0
1936 010422  012702  000002               MOV     #2,R2
1937 010426  022720  177777     1$:       CMP     #-1,(R0)+
1938 010432  001402                       BEQ     2$
1939 010434  000137  010744               JMP     IERR1
1940 010440  077206             2$:       SOB     R2,1$
1941 010442  000401                       BR      I106
1942 010444  077216             I10:      SOB     R2,I7
1943 010446  000137  010764     I106:     JMP     IERR2
1944
1945 010452  077227             I105:     SOB     R2,I6
```

```
1946
1947                                    ;NOW TEST THE LOAD INSTRUCTION WITH FSRC MODE ZERO AND FD CLEAR.
1948
1949 010454                             I11:
     010454  012737  010462  001110              MOV     #I12,$LPERR     ;SET UP THE LOOP ON ERROR ADDRESS.
1950 010462  012700  011062              I12:    MOV     #IPAT10,R0
1951 010466  012701  011112                      MOV     #IDAT10,R1
1952 010472  012702  000004                      MOV     #4,R2
1953 010476  012021                      I13:    MOV     (R0)+,(R1)+
1954 010500  077202                              SOB     R2,I13
1955
1956 010502  012700  011112                      MOV     #IDAT10,R0              ;SET UP AC1
1957 010506  172510                              LDD     (R0),AC1
1958
1959 010510  012700  011072                      MOV     #IPAT20,R0              ;SET UP AC0
1960 010514  172410                              LDD     (R0),AC0
1961
1962 010516  012701  000001                      MOV     #1,R1
1963 010522  012737  010540  001236              MOV     #I14,$TMP2
1964 010530  012737  043576  001240              MOV     #MS36,$TMP3
1965 010536  170001                              SETF                            ;CLEAR FD.
1966
1967 010540  172401                      I14:    LDF     AC1,AC0                 ;TEST INSTRUCTION.
1968 010542  000240                      I15:    NOP
1969 010544  000240                      I16:    NOP
1970
1971 010546  170200                              STFPS   R0                      ;SEE IF FPS IS STILL CLEAR.
1972 010550  022700  000004                      CMP     #4,R0
1973 010554  001402                              BEQ     I17
1974 010556  000137  011036                      JMP     IERR3
1975
1976 010562                              I17:                                    ;RESET TO DOUBLE MODE.
1977 010562  170011                              SETD
1978
1979 010564  012700  011102                      MOV     #IDAT00,R0
1980 010570  174010                              STD     AC0,(R0)                ;GET AC0
1981
1982 010572  012737  177777  011116              MOV     #-1,IDAT12
1983 010600  012737  177777  011120              MOV     #-1,IDAT13
1984 010606  012700  011102                      MOV     #IDAT00,R0
1985 010612  012701  011112                      MOV     #IDAT10,R1
1986 010616  012702  000004                      MOV     #4,R2
1987 010622  022021                      I20:    CMP     (R0)+,(R1)+             ;SEE IF AC0 WAS CORRECT.
1988 010624  001414                              BEQ     I23
1989
1990 010626  023737  011106  011066              CMP     IDAT02,IPAT12           ;DID (BUT FD) FAIL?
1991 010634  001402                              BEQ     I22
1992 010636  000137  010744              I21:    JMP     IERR1
1993 010642  023737  011110  011070      I22:    CMP     IDAT03,IPAT13
1994 010650  001372                              BNE     I21
1995 010652  000137  011012                      JMP     IERR4
1996
1997 010656  077217                      I23:    SOB     R2,I20
1998
1999 010660  000520                              BR      IDONE                   ;NO ERRORS.
2000
2001                                    ;IF AN ODD ADDRESS TRAP OCCURS COME HERE TO ANALYZE THE FSRC FAILURE.
```

```
                                                                    N 4
2002 010662  022716  010350          IERR0:  CMP     #I4,(SP)            ;MAKE SURE THE TRAP OCCURRED
2003 010666  001413                         BEQ     1$                  ;ON THE INSTRUCTION BEING TESTED.
2004 010670  022716  010352                 CMP     #I5,(SP)
2005 010674  001410                         BEQ     1$
2006 010676  022716  010542                 CMP     #I15,(SP)
2007 010702  001405                         BEQ     1$
2008 010704  022716  010544                 CMP     #I16,(SP)
2009 010710  001402                         BEQ     1$
2010 010712  000137  041174                 JMP     CPSPUR
2011
2012 010716  011637  001236          1$:     MOV     (SP),$TMP2          ;REPORT FAILURE.
2013 010722  012737  000627  001240         MOV     #627,$TMP3
2014 010730  012737  000320  001242         MOV     #320,$TMP4
2015 010736  022626                         CMP     (SP)+,(SP)+
2016 010740  104047                 2$:     ERROR   +47
2017 010742  000467                         BR      IDONE
2018
2019                                 ;REPORT DATA ERROR.
2020 010744                          IERR1:
2021 010744  012737  011112  001242         MOV     #IDATI0,$TMP4
2022 010752  012737  011102  001244         MOV     #IDATO0,$TMP5
2023 010760  104051                 1$:     ERROR   +51
2024 010762  000457                         BR      IDONE
2025
2026                                 ;REPORT FAILURE OF (BUT FD)
2027 010764  012737  000153  001244  IERR2:  MOV     #153,$TMP5
2028 010772  012737  000434  001246         MOV     #434,$TMP6
2029 011000  012737  000435  001250         MOV     #435,$TMP7
2030 011006                          IERR25:
2031 011006  104050                 1$:     ERROR   +50
2032 011010  000444                         BR      IDONE
2033 011012  012737  000153  001244  IERR4:  MOV     #153,$TMP5
2034 011020  012737  000435  001246         MOV     #435,$TMP6
2035 011026  012737  000434  001250         MOV     #434,$TMP7
2036 011034  000764                         BR      IERR25
2037
2038                                 ;REPORT INCORRECT FPS AFTER LOAD INSTRUCTION.
2039 011036                          IERR3:
2040 011036  012737  010540  001236         MOV     #I14,$TMP2
2041 011044  010037  001240                 MOV     R0,$TMP3
2042 011050  012737  000004  001242         MOV     #4,$TMP4
2043 011056  104041                 1$:     ERROR   +41
2044 011060  000420                         BR      IDONE
2045
2046
2047 011062  000000                  IPAT10: 0
2048 011064  170360                  IPAT11: 170360
2049 011066  016161                  IPAT12: 016161
2050 011070  052525                  IPAT13: 052525
2051
2052 011072  177777                  IPAT20: -1
2053 011074  177777                  IPAT21: -1
2054 011076  177777                  IPAT22: -1
2055 011100  177777                  IPAT23: -1
2056
2057 011102  000000                  IDAT00: 0
2058 011104  000000                  IDAT01: 0
```

```
2059 011106  000000                IDAT02: 0
2060 011110  000000                IDAT03: 0
2061
2062 011112  000000                IDATI0: 0
2063 011114  000000                IDATI1: 0
2064 011116  000000                IDATI2: 0
2065 011120  000000                IDATI3: 0
2066
2067 011122                        IDONE:
     011122  104413                        RSETUP                  ;GO INITIALIZE THE FPS AND STACK; AND
                                                                    ;SEE IF THE USER HAS EXPRESSED
                                                                    ;THE DESIRE TO CHANGE THE SOFTWARE
                                                                    ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
                                                                    ;THE USER TYPED CONTROL G?).
2068
2074
```

```
                                .SBTTL  TEST # 10 - FDST MODE 0 TEST
                                ;;*****************************************************************
                                ;*TEST 10 - FDST MODE 0 TEST
                                ;*
                                ;*THIS IS A TEST OF THE STORE INSTRUCTIONS, STD AND STF,  WITH FDST MODE 0.
                                ;*
                                ;;*****************************************************************
    011124  000004              TST10:  SCOPE
2076 011126                     T1:
    011126  012737  011134  001110      MOV     #1$,$LPERR      ;SET UP THE LOOP ON ERROR ADDRESS.
2077 011134  170011             1$:     SETD                    ;SET FD
2078 011136  012700  011700             MOV     #TPAT10,R0
2079 011142  012701  011730             MOV     #TDAT10,R1
2080 011146  012702  000004             MOV     #4,R2
2081 011152  012021             T2:     MOV     (R0)+,(R1)+     ;SET UP THE INPUT DATA BUFFER.
2082 011154  077202                     SOB     R2,T2
2083
2084 011156  012700  011730             MOV     #TDAT10,R0      ;LOAD AC0
2085 011162  172410                     LDD     (R0),AC0
2086
2087 011164  012700  011710             MOV     #TPAT20,R0      ;LOAD AC1
2088 011170  172510                     LDD     (R0),AC1
2089
2090 011172  012701  000001             MOV     #1,R1           ;IF THE (BUT FDST) FORK FAILS
2091 011176  012737  011506  000004     MOV     #TERR0,ERRVECT  ;AN ODD ADDRESS TRAP COULD RESULT.
2092 011204  012737  011220  001236     MOV     #T3,$TMP2
2093 011212  012737  043571  001240     MOV     #MS35,$TMP3
2094 011220  174001             T3:     STD     AC0,AC1
2095 011222  000240             T4:     NOP
2096 011224  000240             T5:     NOP
2097
2098 011226  012700  011720             MOV     #TDAT00,R0
2099 011232  174110                     STD     AC1,(R0)        ;GET THE DATA.
2100
2101 011234  012703  011720             MOV     #TDAT00,R3      ;SEE IF THE DATA IS CORRECT.
2102 011240  012704  011730             MOV     #TDAT10,R4
2103 011244  012705  000004             MOV     #4,R5
2104 011250  022324             T6:     CMP     (R3)+,(R4)+
2105 011252  001413                     BEQ     T105
2106
2107 011254  012703  011724             MOV     #TDAT02,R3      ;DID (BUT FD) FAIL?
2108 011260  012705  000002             MOV     #2,R5
2109 011264  005723             T7:     TST     (R3)+
2110 011266  001402                     BEQ     T10
2111 011270  000137  011570             JMP     TERR1
2112 011274  077505             T10:    SOB     R5,T7
2113 011276  000137  011610             JMP     TERR2
2114
2115 011302  077516             T105:   SOB     R5,T6
2116
2117                            ;NOW TEST THE STF AC0,AC1 INSTRUCTION.
2118
2119 011304                     T11:
    011304  012737  011312  001110      MOV     #T12,$LPERR     ;SET UP THE LOOP ON ERROR ADDRESS.
2120
2121 011312  012700  011700     T12:    MOV     #TPAT10,R0      ;SET UP THE INPUT DATA BUFFER.
2122 011316  012701  011730             MOV     #TDAT10,R1
```

```
2123 011322  012702  000004              MOV     #4,R2
2124 011326  012021              T13:    MOV     (R0)+,(R1)+
2125 011330  077202                      SOB     R2,T13
2126
2127 011332  012700  011730              MOV     #TDATI0,R0              ;SET UP AC0
2128 011336  172410                      LDD     (R0),AC0
2129
2130 011340  012700  011710              MOV     #TPAT20,R0              ;SET UP AC1
2131 011344  172510                      LDD     (R0),AC1
2132
2133 011346  012701  000001              MOV     #1,R1
2134 011352  012737  011370  001236      MOV     #T14,$TMP2
2135 011360  012737  043576  001240      MOV     #MS36,$TMP3
2136 011366  170001                      SETF                            ;CLEAR FD
2137 011370  174001              T14:    STF     AC0,AC1
2138 011372  000240              T15:    NOP
2139 011374  000240              T16:    NOP
2140
2141 011376  005000                      CLR     R0
2142 011400  170200                      STFPS   R0                      ;SEE IF FPS IS CLEAR.
2143 011402  022700  000010              CMP     #10,R0
2144 011406  001401                      BEQ     T17
2145 011410  000521                      BR      TERR3
2146
2147 011412              T17:
2148 011412  170011                      SETD                            ;SET FD.
2149
2150 011414  012700  011720              MOV     #TDATO0,R0
2151 011420  174110                      STD     AC1,(R0)                ;PICK UP AC1.
2152
2153 011422  012737  177777  011734      MOV     #-1,TDATI2
2154 011430  012737  177777  011736      MOV     #-1,TDATI3
2155 011436  012703  011720              MOV     #TDATO0,R3
2156 011442  012704  011730              MOV     #TDATI0,R4
2157 011446  012705  000004              MOV     #4,R5
2158 011452  022324              T20:    CMP     (R3)+,(R4)+             ;WAS THE DATA TRANSFERRED CORRECTLY?
2159 011454  001412                      BEQ     T23
2160
2161 011456  023737  011724  011704      CMP     TDATO2,TPAT12           ;DID (BUT FD) FAIL.
2162 011464  001401                      BEQ     T22
2163 011466  000440              T21:    BR      TERR1
2164 011470  023737  011726  011706 T22: CMP     TDATO3,TPAT13
2165 011476  001373                      BNE     T21
2166 011500  000456                      BR      TERR4
2167
2168 011502  077515              T23:    SOB     R5,T20
2169 011504  000515                      BR      TDONE
2170
2171
2172                          ;TRAP HERE THROUGH VECTOR 4 IF AN ODD ADDRESS OCCURS.
2173 011506  022716  011222  TERR0: CMP     #T4,(SP)                ;MAKE SURE THE TRAP WAS ON
2174 011512  001413                      BEQ     1$                      ;AN INSTRUCTION BEING TESTED.
2175 011514  022716  011224              CMP     #T5,(SP)
2176 011520  001410                      BEQ     1$
2177 011522  022716  011372              CMP     #T15,(SP)
2178 011526  001405                      BEQ     1$
2179 011530  022716  011374              CMP     #T16,(SP)
```

```
011534  001402                            BEQ     1$
011536  000137  041174                    JMP     CPSPUR

011542  011637  001236            1$:     MOV     (SP),$TMP2
011546  022626                            CMP     (SP)+,(SP)+
011550  012737  000527  001240            MOV     #527,$TMP3
011556  012737  000640  001242            MOV     #640,$TMP4
011564  104121                    2$:     ERROR   +121
011566  000464                            BR      TDONE

                                  ;REPORT DATA FAILURE.
011570                            TERR1:
011570  012737  011730  001242            MOV     #TDATI0,$TMP4
011576  012737  011720  001244            MOV     #TDAT00,$TMP5
011604  104123                    1$:     ERROR   +123
011606  000454                            BR      TDONE

                                  ;REPORT FAILURE OF (BUT FD).
011610  012737  000160  001246    TERR2:  MOV     #160,$TMP6
011616  012737  000161  001250            MOV     #161,$TMP7
011624  012737  000640  001244    TERR25: MOV     #640,$TMP5
011632  104122                    1$:     ERROR   +122
011634  000441                            BR      TDONE
011636  012737  000161  001246    TERR4:  MOV     #161,$TMP6
011644  012737  000160  001250            MOV     #160,$TMP7
011652  000764                            BR      TERR25

                                  ;REPORT INCORRECT FPS AFTER STORE INSTRUCTION.
011654                            TERR3:
011654  012737  011372  001236            MOV     #T15,$TMP2
011662  010037  001240                    MOV     R0,$TMP3
011666  012737  000010  001242            MOV     #10,$TMP4
011674  104041                    1$:     ERROR   +41
011676  000420                            BR      TDONE

011700  000000                    TPAT10: 0
011702  170360                    TPAT11: 170360
011704  016161                    TPAT12: 016161
011706  052525                    TPAT13: 052525

011710  177777                    TPAT20: -1
011712  177777                    TPAT21: -1
011714  177777                    TPAT22: -1
011716  177777                    TPAT23: -1

011720  000000                    TDAT00: 0
011722  000000                    TDAT01: 0
011724  000000                    TDAT02: 0
011726  000000                    TDAT03: 0

011730  000000                    TDATI0: 0
011732  000000                    TDATI1: 0
011734  000000                    TDATI2: 0
011736  000000                    TDATI3: 0

011740                            TDONE:
011740  104413                            RSETUP                          ;GO INITIALIZE THE FPS AND STACK; AND
```

```
                                        ;SEE IF THE USER HAS EXPRESSED
                                        ;THE DESIRE TO CHANGE THE SOFTWARE
                                        ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
                                        ;THE USER TYPED CONTROL G?).

2236
2237
2371
```

```
2372                    .SBTTL  TEST # 11 - ACCUMULATORS DATA PATTERNS TEST
                        ;;**********************************************************************
                        ;*TEST 11 - ACCUMULATORS DATA PATTERNS TEST
                        ;*
                        ;*THIS IS A TEST OF THE FLOATING POINT PROCESSOR ACCUMULATORS.
                        ;*EACH ACCUMULATOR IS TESTED IN TWO WAYS:
                        ;*      1 TEST PATTERN GENERATED BY FLOATING A ONE ACROSS
                        ;*        A FIELD OF ZEROES.
                        ;*      2 TEST PATTERN GENERATED BY FLOATING A ZERO ACROSS
                        ;*        A FIELD OF ONES.
                        ;*EACH OF ACCUMULATORS AC0 THROUGH AC5 IS TESTED.
                        ;*
                        ;*NOTE THAT THIS TEST KEEPS A DYNAMIC RECORD OF THE LOGICAL 'AND' AND 'OR'
                        ;*OF THE FAILING DATA PATTERNS. THESE CAN BE VERY USEFUL IN DETERMINING
                        ;*STUCK BITS. IF THE USER HAS THE INHIBIT ERROR TYPE OUT SWITCH (SWR13)
                        ;*OFF, THEN THE USER WILL RECIEVE EACH INDIVIDUAL ERROR MESSAGE PLUS
                        ;*AN ERROR SUMMARY AT THE END OF THE TEST. INHIBITING ERROR PRINT OUT
                        ;*WILL INHIBIT ERROR SUMMARY PRINT OUT, EXCEPT IN THE CASE DESCRIBED BELOW.
                        ;*TO GET JUST THE ERROR SUMMARY WITH NO INDIVIDUAL ERROR REPORTS,
                        ;*SET SWITCH REGISTER BIT13 AND SWITCH REGISTER BIT7 BOTH ON.
                        ;*
                        ;*
                        ;*THE FOLLOWING PROCEDURE IS PRESENTED TO AID THE TROUBLE
                        ;*SHOOTER IN SITUATIONS WHERE AM2901 CHIP ISOLATION IS ATTEMPTED.
                        ;*
                        ;*WARNING: THIS PROCEDURE ASSUMES THAT THE FAULT IS IN ONE OF THE
                        ;*AM2901 CHIPS. THIS ASSUMPTION IS NOT NECESSARILY VALID IN ALL
                        ;*SITUATIONS. IT REMAINS TO BE SEEN WHAT NUMBER OF FAILURES CAN
                        ;*PROBABLILISTICALLY ASSOCIATED WITH THEM. NOTE ALSO THAT THIS
                        ;*INFORMATION SHOULD NOT BE TAKEN AS ABSOLUTE, THAT IS
                        ;*THIS INFORMATION IS THE AUTHOR'S SUGGESTION FOR ACHIEVING ISOLATION
                        ;*WHEN CHIP LEVEL REPAIR IS NECESSARY.
                        ;*
                        ;*WHEN THIS TEST HAS FINISHED RUNNING, IF ERRORS HAVE OCCURRED,
                        ;*AN ERROR SUMMARY WILL BE TYPED. THUS SUMMARY WILL CONSIST OF TWO
                        ;*IMPORTANT QUANTITIES:
                        ;*      A.      FOUR SIXTEEN BIT NUMBERS LABELED THE LOGICAL 'AND' ('*')
                        ;*              OF THE FAILING DATA PATTERNS.
                        ;*      B.      FOUR SIXTEEN BIT NUMBERS LABELED THE LOGICAL 'OR' ('+')
                        ;*              OF THE FAILING DATA PATTERNS.
                        ;*
                        ;*A BIT STUCK HIGH IN THE HARDWARE WILL SHOW UP AS A 0 IN THAT
                        ;*BIT POSITION OF THE 'OR' OF THE FAILING DATA PATTERNS.
                        ;*
                        ;*A BIT STUCK LOW IN THE HARDWARE WILL SHOW UP AS A 1 IN THAT BIT
                        ;*POSITION OF THE 'AND' OF THE FAILING DATA PATTERNS.
                        ;*
                        ;*THUS IF A FAILURE OCCURS:
                        ;*      A.      STUCK HIGHS WILL SHOW AS 0'S IN THE 'OR' PATTERN.
                        ;*      B.      STUCK LOWS WILL SHOW AS 1'S IN THE 'AND' PATTERN.
                        ;*IF THE FAILURE IS INTERMITTANT THEN THIS PROCEDURE WILL STILL
                        ;*APPLY!!.
                        ;*IF THE FAILURE MOVES FROM ONE BIT TO ANOTHER, OR FROM ONE
                        ;*GROUP OF BITS TO ANOTHER GROUP OF BITS THEN THE FAULT WILL
                        ;*PROBABLY NOT SHOW UP IN THE 'AND' OR THE 'OR' PATTERNS; IN THIS
                        ;*CASE THE 'AND' PATTERN WILL BE ALL 0'S AND THE 'OR' PATTERN WILL
                        ;*BE ALL 1'S. WHEN THIS OCCURS SOME OTHER METHOD OF REPAIR MUST
```

```
;*BE FOUND (SUCH AS INSPECTION OF EACH INDIVIDUAL ERROR REPORT
;*RATHER THAN USING THE SUMMARY).
;*
;*MAP THE FOLLOWING NOTATION ONTO EACH BIT POSITION IN THE 'AND'
;*AND THE 'OR' PATTERNS WHICH ARE TYPED IN THE ERROR SUMMARY.
;*
;*A15,A14,...A1,A0  B15,B14,...B1,B0  C15,C14,...C1,C0  D15,D14,...C1,C0
;*
;*IN THIS NOTATION A15 THROUGH A0 IS THE FIRST OF THE FOUR 16 BIT
;*OCTAL NUMBERS TYPED, B15 THROUGH B0 IS THE SECOND, ETC.
;*
;*THIS TABLE SHOWS THE CORRESPONDING AM2901 CHIP ('E' NUMBER)
;*WHICH IS RESPONSIBLE FOR EACH BIT POSITION USING THE ABOVE
;*NOTATION. NOTE THAT ECO'S TO THE HARDWARE MIGHT MAKE THIS
;*TABLE OBSOLETE IF IT IS NOT UP DATED. NOTE ALSO THAT THERE ARE
;*FOUR BITS FOR EACH AM2901 CHIP:
;*
;*       BITS                          AM2901 CHIP NUMBER
;*       ----                          ------------------
;*
;*       A15,A14,A13,A12               E37
;*       A11,A10,A9,A8                 E45
;*       A7,A6,A5,A4                   E34
;*       A3,A2,A1,A0                   E42
;*
;*       B15,B14,B13,B12               E33
;*       B11,B10,B9,B8                 E41
;*       B7,B6,B5,B4                   E36
;*       B3,B2,B1,B0                   E44
;*
;*       C15,C14,C13,C12               E35
;*       C11,C10,C9,C8                 E43
;*       C7,C6,C5,C4                   E38
;*       C3,C2,C1,C0                   E46
;*
;*       D15,D14,D13,D12               E39
;*       D11,D10,D9,D8                 E47
;*       D7,D6,D5,D4                   E40
;*       D3,D2,D1,D0                   E48
;*
;*NOW FIVE IMPORTANT CASES WHICH WILL ARISE WHEN A FAULTY
;*AM2901 IS PRESENT CAN BE DESCRIBED:
;*
;*1.)    IF ONLY ONE BIT OF THE 64 BITS IS INCORRECT THE CHIP INDICATED
;*       IN THE ABOVE TABLE IS MOST PROBABLY AT FAULT. BUT IF THAT
;*       CHIP IS REPLACED AND THE ERROR PERSISTS THEN SUPPOSE THAT
;*       BIT IS,
;*              LN                WHERE 'L' IS A, B, C OR D
;*                                ANDN IS 15, 14, ... OR 0
;*       THEN IN GENERAL ANY OF THE FOUR CHIPS RESPONSIBLE FOR
;*       AN, BN, CN OR DN COULD BE AT FAULT, WITH LN BEING MUST PROBABLE.
;*       FOR EXAMPLE IF BIT C12 IS FAULTY, THEN CHIP E79
;*       IS THE MOST PROBABLE SOURCE OF THE ERROR. IF REPAIRING
;*       THAT CHIP DOES NOT REMOVE THE FAULT THEN TRY EACH OF THE
;*       CHIPS ASSOCIATED WITH BITS A12, B12 AND D12 SHOULD BE TRIED
;*       WITH EQUAL PROBABILITY OF THE FAULT BEING
;*       IN ANY ONE OF THESE OTHER THREE CHIPS, TRY CHIPS E61, E86 AND E78.
```

```
                        ;*2.)   IF THERE ARE FOUR CONSECUTIVE BITS IN ERROR, FOLLOWING THE
                        ;*      PATTERN:
                        ;*              LN, LN+1, LN+2 AND LN+3          WHERE 'L' IS A, B, C
                        ;*                                               OR D.
                        ;*                                               AND N=0,4,8 OR 12
                        ;*      THEN THE ABOVE TABLE SHOULD DIRECTLY IDENTIFY THE
                        ;*      FAILING CHIP.
                        ;*
                        ;*3.)   IF FOUR BITS ARE DROPPED WHICH FIT THE PATTERN:
                        ;*              AN, BN, CN AND DN        WHERE N=15,14,... OR 0
                        ;*      THEN ANY ONE OF THE FOUR CHIPS ASSOCIATED WITH EACH OF
                        ;*      THE BITS AN, BN, CN AND DN COULD BE AT FAULT WITH
                        ;*      EQUAL PROBABILITY.
                        ;*
                        ;*4.)   IF 16 BITS ARE IN ERROR, FITTING THE PATTERN:
                        ;*              AN, AN+1, AN+2, AN+3     WHERE N=0,4,8 OR 12
                        ;*              BN, BN+1, BN+2, BN+3
                        ;*              CN, CN+1, CN+2, CN+3
                        ;*                      AND
                        ;*              DN, DN+1, DN+2, AN+3
                        ;*      THEN ANY ONE OF THE FOUR CHIPS ASSOCIATED
                        ;*      WITH THESE BITS COULD BE AT FAULT WITH EQUAL PROBABILITY.
                        ;*
                        ;*5.)   IF THE FAILING BIT PATTERNS DISPLAYED IN THE 'AND' AND THE 'OR'
                        ;*      DATA TYPED IN THE SUMMARY DOES NOT CONFORM EXPLICITELY TO
                        ;*      ANY OF THE ABOVE PATTERNS, THEN THE TROUBLE SHOOTER MUST
                        ;*      INTUITIVELY TRY TO FIND WHICH OF THE ABOVE CASES (1 THROUGH 4)
                        ;*      IS A 'BEST FIT' OF THE SYMPTOMS.
                        ;*
                        ;;*****************************************************************
 011742  000004          TST11:  SCOPE
2373 011744  170011              SETD                            ;SET FD.
2374
                        ;TEST ACCUMULATOR 0 WITH FLOATING ONE
 011746  012737  043143  001244          MOV     #MNUM0,$TMP5
 011754  012737  012010  001236          MOV     #G1,$TMP2
 011762  012700  014300                  MOV     #GPAT00,R0
 011766  012701  014340                  MOV     #GDAT00,R1
 011772  012737  012010  001110          MOV     #G1,$LPERR      ;SET UP THE LOOP ON ERROR ADDRESS.
 012000  004737  013762                  JSR     PC,GSETUP               ;LOAD TEST PATTERN.
 012004  012703  000102                  MOV     #102,R3
 012010                          G1:
 012010  172410                          LDD     (R0),AC0
 012012  174000                          STD     AC0,AC0
 012014  172400                          LDD     AC0,AC0         ;STORE THE TEST PATTERN.
 012016  174011                          STD     AC0,(R1)
 012020  004737  014060                  JSR     PC,GCMP                 ;COMPARE THE DATA READ WITH
                                                                 ;THAT WHICH WAS WRITTEN.
 012024  005737  014274                  TST     GFLAG1
 012030  001004                          BNE     G2
 012032  005137  014274                  COM     GFLAG1
 012036  000261                          SEC
 012040  000401                          BR      G3
 012042  000241                  G2:     CLC
 012044  006160  000006          G3:     ROL     6(R0)                   ;GENERATE THE NEXT TEST PATTERN.
 012050  006160  000004                  ROL     4(R0)
 012054  006160  000002                  ROL     2(R0)
 012060  006110                          ROL     (R0)
```

```
        012062  004737  014040                 JSR     PC,GRESET               ;RESET DEFAULT PATTERN IN OUTPUT
                                                                                ;BUFFER.
        012066  077330                          SOB     R3,G1
        012070  004737  014172                  JSR     PC,GSUM                 ;TYPE ERROR SUMMARY.
  2375                          ;TEST ACCUMULATOR 0 WITH FLOATING ZERO
        012074  012737  043143  001244          MOV     #MNUM0,$TMP5
        012102  012737  012136  001236          MOV     #G4,$TMP2
        012110  012700  014310                  MOV     #GPAT10,R0
        012114  012701  014340                  MOV     #GDAT00,R1
        012120  012737  012136  001110          MOV     #G4,$LPERR      ;SET UP THE LOOP ON ERROR ADDRESS.
        012126  004737  013762                  JSR     PC,GSETUP               ;LOAD TEST PATTERN.
        012132  012703  000102                  MOV     #102,R3
        012136                          G4:
        012136  172410                          LDD     (R0),AC0
        012140  174000                          STD     AC0,AC0
        012142  172400                          LDD     AC0,AC0         ;STORE THE TEST PATTERN.
        012144  174011                          STD     AC0,(R1)
        012146  004737  014060                  JSR     PC,GCMP                 ;COMPARE THE DATA READ WITH
                                                                                ;THAT WHICH WAS WRITTEN.
        012152  005737  014274                  TST     GFLAG1
        012156  001004                          BNE     G5
        012160  005137  014274                  COM     GFLAG1
        012164  000241                          CLC
        012166  000401                          BR      G6
        012170  000261                  G5:     SEC
        012172  006160  000006          G6:     ROL     6(R0)                   ;GENERATE THE NEXT TEST PATTERN.
        012176  006160  000004                  ROL     4(R0)
        012202  006160  000002                  ROL     2(R0)
        012206  006110                          ROL     (R0)
        012210  004737  014040                  JSR     PC,GRESET               ;RESET DEFAULT PATTERN IN OUTPUT
                                                                                ;BUFFER.
        012214  077330                          SOB     R3,G4
        012216  004737  014172                  JSR     PC,GSUM                 ;TYPE ERROR SUMMARY.
  2376                          ;TEST ACCUMULATOR 1 WITH FLOATING ONE
        012222  012737  043151  001244          MOV     #MNUM1,$TMP5
        012230  012737  012264  001236          MOV     #G7,$TMP2
        012236  012700  014300                  MOV     #GPAT00,R0
        012242  012701  014340                  MOV     #GDAT00,R1
        012246  012737  012264  001110          MOV     #G7,$LPERR      ;SET UP THE LOOP ON ERROR ADDRESS.
        012254  004737  013762                  JSR     PC,GSETUP               ;LOAD TEST PATTERN.
        012260  012703  000102                  MOV     #102,R3
        012264                          G7:
        012264  172410                          LDD     (R0),AC0
        012266  174001                          STD     AC0,AC1
        012270  172401                          LDD     AC1,AC0         ;STORE THE TEST PATTERN.
        012272  174011                          STD     AC0,(R1)
        012274  004737  014060                  JSR     PC,GCMP                 ;COMPARE THE DATA READ WITH
                                                                                ;THAT WHICH WAS WRITTEN.
        012300  005737  014274                  TST     GFLAG1
        012304  001004                          BNE     G10
        012306  005137  014274                  COM     GFLAG1
        012312  000261                          SEC
        012314  000401                          BR      G11
        012316  000241                  G10:    CLC
        012320  006160  000006          G11:    ROL     6(R0)                   ;GENERATE THE NEXT TEST PATTERN.
        012324  006160  000004                  ROL     4(R0)
        012330  006160  000002                  ROL     2(R0)
```

```
        012334  006110                          ROL     (R0)
        012336  004737  014040                  JSR     PC,GRESET               ;RESET DEFAULT PATTERN IN OUTPUT
                                                                                ;BUFFER.
        012342  077330                          SOB     R3,G7
        012344  004737  014172                  JSR     PC,GSUM                 ;TYPE ERROR SUMMARY.
  2377                          ;TEST ACCUMULATOR 1 WITH FLOATING ZERO
        012350  012737  043151  001244          MOV     #MNUM1,$TMP5
        012356  012737  012412  001236          MOV     #G12,$TMP2
        012364  012700  014310                  MOV     #GPAT10,R0
        012370  012701  014340                  MOV     #GDAT00,R1
        012374  012737  012412  001110          MOV     #G12,$LPERR     ;SET UP THE LOOP ON ERROR ADDRESS.
        012402  004737  013762                  JSR     PC,GSETUP               ;LOAD TEST PATTERN.
        012406  012703  000102                  MOV     #102,R3
        012412                          G12:
        012412  172410                          LDD     (R0),AC0
        012414  174001                          STD     AC0,AC1
        012416  172401                          LDD     AC1,AC0         ;STORE THE TEST PATTERN.
        012420  174011                          STD     AC0,(R1)
        012422  004737  014060                  JSR     PC,GCMP                 ;COMPARE THE DATA READ WITH
                                                                                ;THAT WHICH WAS WRITTEN.
        012426  005737  014274                  TST     GFLAG1
        012432  001004                          BNE     G13
        012434  005137  014274                  COM     GFLAG1
        012440  000241                          CLC
        012442  000401                          BR      G14
        012444  000261                  G13:    SEC
        012446  006160  000006          G14:    ROL     6(R0)                   ;GENERATE THE NEXT TEST PATTERN.
        012452  006160  000004                  ROL     4(R0)
        012456  006160  000002                  ROL     2(R0)
        012462  006110                          ROL     (R0)
        012464  004737  014040                  JSR     PC,GRESET               ;RESET DEFAULT PATTERN IN OUTPUT
                                                                                ;BUFFER.
        012470  077330                          SOB     R3,G12
        012472  004737  014172                  JSR     PC,GSUM                 ;TYPE ERROR SUMMARY.
  2378                          ;TEST ACCUMULATOR 2 WITH FLOATING ONE
        012476  012737  043156  001244          MOV     #MNUM2,$TMP5
        012504  012737  012540  001236          MOV     #G15,$TMP2
        012512  012700  014300                  MOV     #GPAT00,R0
        012516  012701  014340                  MOV     #GDAT00,R1
        012522  012737  012540  001110          MOV     #G15,$LPERR     ;SET UP THE LOOP ON ERROR ADDRESS.
        012530  004737  013762                  JSR     PC,GSETUP               ;LOAD TEST PATTERN.
        012534  012703  000102                  MOV     #102,R3
        012540                          G15:
        012540  172410                          LDD     (R0),AC0
        012542  174002                          STD     AC0,AC2
        012544  172402                          LDD     AC2,AC0         ;STORE THE TEST PATTERN.
        012546  174011                          STD     AC0,(R1)
        012550  004737  014060                  JSR     PC,GCMP                 ;COMPARE THE DATA READ WITH
                                                                                ;THAT WHICH WAS WRITTEN.
        012554  005737  014274                  TST     GFLAG1
        012560  001004                          BNE     G16
        012562  005137  014274                  COM     GFLAG1
        012566  000261                          SEC
        012570  000401                          BR      G17
        012572  000241                  G16:    CLC
        012574  006160  000006          G17:    ROL     6(R0)                   ;GENERATE THE NEXT TEST PATTERN.
        012600  006160  000004                  ROL     4(R0)
```

```
        012604  006160  000002                  ROL     2(R0)
        012610  006110                          ROL     (R0)
        012612  004737  014040                  JSR     PC,GRESET               ;RESET DEFAULT PATTERN IN OUTPUT
                                                                                ;BUFFER.
        012616  077330                          SOB     R3,G15
        012620  004737  014172                  JSR     PC,GSUM                 ;TYPE ERROR SUMMARY.
  2379                          ;TEST ACCUMULATOR 2 WITH FLOATING ZERO
        012624  012737  043156  001244          MOV     #MNUM2,$TMP5
        012632  012737  012666  001236          MOV     #G20,$TMP2
        012640  012700  014310                  MOV     #GPAT10,R0
        012644  012701  014340                  MOV     #GDAT00,R1
        012650  012737  012666  001110          MOV     #G20,$LPERR     ;SET UP THE LOOP ON ERROR ADDRESS.
        012656  004737  013762                  JSR     PC,GSETUP               ;LOAD TEST PATTERN.
        012662  012703  000102                  MOV     #102,R3
        012666                          G20:
        012666  172410                          LDD     (R0),AC0
        012670  174002                          STD     AC0,AC2
        012672  172402                          LDD     AC2,AC0         ;STORE THE TEST PATTERN.
        012674  174011                          STD     AC0,(R1)
        012676  004737  014060                  JSR     PC,GCMP                 ;COMPARE THE DATA READ WITH
                                                                                ;THAT WHICH WAS WRITTEN.
        012702  005737  014274                  TST     GFLAG1
        012706  001004                          BNE     G21
        012710  005137  014274                  COM     GFLAG1
        012714  000241                          CLC
        012716  000401                          BR      G22
        012720  000261                  G21:    SEC
        012722  006160  000006          G22:    ROL     6(R0)                   ;GENERATE THE NEXT TEST PATTERN.
        012726  006160  000004                  ROL     4(R0)
        012732  006160  000002                  ROL     2(R0)
        012736  006110                          ROL     (R0)
        012740  004737  014040                  JSR     PC,GRESET               ;RESET DEFAULT PATTERN IN OUTPUT
                                                                                ;BUFFER.
        012744  077330                          SOB     R3,G20
        012746  004737  014172                  JSR     PC,GSUM                 ;TYPE ERROR SUMMARY.
  2380                          ;TEST ACCUMULATOR 3 WITH FLOATING ONE
        012752  012737  043163  001244          MOV     #MNUM3,$TMP5
        012760  012737  013014  001236          MOV     #G23,$TMP2
        012766  012700  014300                  MOV     #GPAT00,R0
        012772  012701  014340                  MOV     #GDAT00,R1
        012776  012737  013014  001110          MOV     #G23,$LPERR     ;SET UP THE LOOP ON ERROR ADDRESS.
        013004  004737  013762                  JSR     PC,GSETUP               ;LOAD TEST PATTERN.
        013010  012703  000102                  MOV     #102,R3
        013014                          G23:
        013014  172410                          LDD     (R0),AC0
        013016  174003                          STD     AC0,AC3
        013020  172403                          LDD     AC3,AC0         ;STORE THE TEST PATTERN.
        013022  174011                          STD     AC0,(R1)
        013024  004737  014060                  JSR     PC,GCMP                 ;COMPARE THE DATA READ WITH
                                                                                ;THAT WHICH WAS WRITTEN.
        013030  005737  014274                  TST     GFLAG1
        013034  001004                          BNE     G24
        013036  005137  014274                  COM     GFLAG1
        013042  000261                          SEC
        013044  000401                          BR      G25
        013046  000241                  G24:    CLC
        013050  006160  000006          G25:    ROL     6(R0)                   ;GENERATE THE NEXT TEST PATTERN.
```

```
013054  006160  000004                  ROL     4(R0)
013060  006160  000002                  ROL     2(R0)
013064  006110                          ROL     (R0)
013066  004737  014040                  JSR     PC,GRESET               ;RESET DEFAULT PATTERN IN OUTPUT
                                                                        ;BUFFER.
013072  077330                          SOB     R3,G23
013074  004737  014172                  JSR     PC,GSUM                 ;TYPE ERROR SUMMARY.
2381                            ;TEST ACCUMULATOR 3 WITH FLOATING ZERO
013100  012737  043163  001244          MOV     #MNUM3,$TMP5
013106  012737  013142  001236          MOV     #G26,$TMP2
013114  012700  014310                  MOV     #GPAT10,R0
013120  012701  014340                  MOV     #GDAT00,R1
013124  012737  013142  001110          MOV     #G26,$LPERR     ;SET UP THE LOOP ON ERROR ADDRESS.
013132  004737  013762                  JSR     PC,GSETUP               ;LOAD TEST PATTERN.
013136  012703  000102                  MOV     #102,R3
013142                          G26:
013142  172410                          LDD     (R0),AC0
013144  174003                          STD     AC0,AC3
013146  172403                          LDD     AC3,AC0         ;STORE THE TEST PATTERN.
013150  174011                          STD     AC0,(R1)
013152  004737  014060                  JSR     PC,GCMP                 ;COMPARE THE DATA READ WITH
                                                                        ;THAT WHICH WAS WRITTEN.
013156  005737  014274                  TST     GFLAG1
013162  001004                          BNE     G27
013164  005137  014274                  COM     GFLAG1
013170  000241                          CLC
013172  000401                          BR      G30
013174  000261                  G27:    SEC
013176  006160  000006          G30:    ROL     6(R0)                   ;GENERATE THE NEXT TEST PATTERN.
013202  006160  000004                  ROL     4(R0)
013206  006160  000002                  ROL     2(R0)
013212  006110                          ROL     (R0)
013214  004737  014040                  JSR     PC,GRESET               ;RESET DEFAULT PATTERN IN OUTPUT
                                                                        ;BUFFER.
013220  077330                          SOB     R3,G26
013222  004737  014172                  JSR     PC,GSUM                 ;TYPE ERROR SUMMARY.
2382                            ;TEST ACCUMULATOR 4 WITH FLOATING ONE
013226  012737  043172  001244          MOV     #MNUM4,$TMP5
013234  012737  013270  001236          MOV     #G31,$TMP2
013242  012700  014300                  MOV     #GPAT00,R0
013246  012701  014340                  MOV     #GDAT00,R1
013252  012737  013270  001110          MOV     #G31,$LPERR     ;SET UP THE LOOP ON ERROR ADDRESS.
013260  004737  013762                  JSR     PC,GSETUP               ;LOAD TEST PATTERN.
013264  012703  000102                  MOV     #102,R3
013270                          G31:
013270  172410                          LDD     (R0),AC0
013272  174004                          STD     AC0,AC4
013274  172404                          LDD     AC4,AC0         ;STORE THE TEST PATTERN.
013276  174011                          STD     AC0,(R1)
013300  004737  014060                  JSR     PC,GCMP                 ;COMPARE THE DATA READ WITH
                                                                        ;THAT WHICH WAS WRITTEN.
013304  005737  014274                  TST     GFLAG1
013310  001004                          BNE     G32
013312  005137  014274                  COM     GFLAG1
013316  000261                          SEC
013320  000401                          BR      G33
013322  000241                  G32:    CLC
```

```
013324  006160  000006          G33:    ROL     6(R0)                   ;GENERATE THE NEXT TEST PATTERN.
013330  006160  000004                  ROL     4(R0)
013334  006160  000002                  ROL     2(R0)
013340  006110                          ROL     (R0)
013342  004737  014040                  JSR     PC,GRESET               ;RESET DEFAULT PATTERN IN OUTPUT
                                                                        ;BUFFER.
013346  077330                          SOB     R3,G31
013350  004737  014172                  JSR     PC,GSUM                 ;TYPE ERROR SUMMARY.
2383                            ;TEST ACCUMULATOR 4 WITH FLOATING ZERO
013354  012737  043172  001244          MOV     #MNUM4,$TMP5
013362  012737  013416  001236          MOV     #G34,$TMP2
013370  012700  014310                  MOV     #GPAT10,R0
013374  012701  014340                  MOV     #GDAT00,R1
013400  012737  013416  001110          MOV     #G34,$LPERR     ;SET UP THE LOOP ON ERROR ADDRESS.
013406  004737  013762                  JSR     PC,GSETUP               ;LOAD TEST PATTERN.
013412  012703  000102                  MOV     #102,R3
013416                          G34:
013416  172410                          LDD     (R0),AC0
013420  174004                          STD     AC0,AC4
013422  172404                          LDD     AC4,AC0         ;STORE THE TEST PATTERN.
013424  174011                          STD     AC0,(R1)
013426  004737  014060                  JSR     PC,GCMP                 ;COMPARE THE DATA READ WITH
                                                                        ;THAT WHICH WAS WRITTEN.
013432  005737  014274                  TST     GFLAG1
013436  001004                          BNE     G35
013440  005137  014274                  COM     GFLAG1
013444  000241                          CLC
013446  000401                          BR      G36
013450  000261          G35:            SEC
013452  006160  000006          G36:    ROL     6(R0)                   ;GENERATE THE NEXT TEST PATTERN.
013456  006160  000004                  ROL     4(R0)
013462  006160  000002                  ROL     2(R0)
013466  006110                          ROL     (R0)
013470  004737  014040                  JSR     PC,GRESET               ;RESET DEFAULT PATTERN IN OUTPUT
                                                                        ;BUFFER.
013474  077330                          SOB     R3,G34
013476  004737  014172                  JSR     PC,GSUM                 ;TYPE ERROR SUMMARY.
2384                            ;TEST ACCUMULATOR 5 WITH FLOATING ONE
013502  012737  043200  001244          MOV     #MNUM5,$TMP5
013510  012737  013544  001236          MOV     #G37,$TMP2
013516  012700  014300                  MOV     #GPAT00,R0
013522  012701  014340                  MOV     #GDAT00,R1
013526  012737  013544  001110          MOV     #G37,$LPERR     ;SET UP THE LOOP ON ERROR ADDRESS.
013534  004737  013762                  JSR     PC,GSETUP               ;LOAD TEST PATTERN.
013540  012703  000102                  MOV     #102,R3
013544                          G37:
013544  172410                          LDD     (R0),AC0
013546  174005                          STD     AC0,AC5
013550  172405                          LDD     AC5,AC0         ;STORE THE TEST PATTERN.
013552  174011                          STD     AC0,(R1)
013554  004737  014060                  JSR     PC,GCMP                 ;COMPARE THE DATA READ WITH
                                                                        ;THAT WHICH WAS WRITTEN.
013560  005737  014274                  TST     GFLAG1
013564  001004                          BNE     G40
013566  005137  014274                  COM     GFLAG1
013572  000261                          SEC
013574  000401                          BR      G41
```

```
013576  000241                  G40:    CLC
013600  006160  000006          G41:    ROL     6(R0)                   ;GENERATE THE NEXT TEST PATTERN.
013604  006160  000004                  ROL     4(R0)
013610  006160  000002                  ROL     2(R0)
013614  006110                          ROL     (R0)
013616  004737  014040                  JSR     PC,GRESET               ;RESET DEFAULT PATTERN IN OUTPUT
                                                                        ;BUFFER.
013622  077330                          SOB     R3,G37
013624  004737  014172                  JSR     PC,GSUM                 ;TYPE ERROR SUMMARY.
2385                            ;TEST ACCUMULATOR 5 WITH FLOATING ZERO
013630  012737  043200  001244          MOV     #MNUM5,$TMP5
013636  012737  013672  001236          MOV     #G42,$TMP2
013644  012700  014310                  MOV     #GPAT10,R0
013650  012701  014340                  MOV     #GDAT00,R1
013654  012737  013672  001110          MOV     #G42,$LPERR     ;SET UP THE LOOP ON ERROR ADDRESS.
013662  004737  013762                  JSR     PC,GSETUP               ;LOAD TEST PATTERN.
013666  012703  000102                  MOV     #102,R3
013672                          G42:
013672  172410                          LDD     (R0),AC0
013674  174005                          STD     AC0,AC5
013676  172405                          LDD     AC5,AC0         ;STORE THE TEST PATTERN.
013700  174011                          STD     AC0,(R1)
013702  004737  014060                  JSR     PC,GCMP                 ;COMPARE THE DATA READ WITH
                                                                        ;THAT WHICH WAS WRITTEN.
013706  005737  014274                  TST     GFLAG1
013712  001004                          BNE     G43
013714  005137  014274                  COM     GFLAG1
013720  000241                          CLC
013722  000401                          BR      G44
013724  000261                  G43:    SEC
013726  006160  000006          G44:    ROL     6(R0)                   ;GENERATE THE NEXT TEST PATTERN.
013732  006160  000004                  ROL     4(R0)
013736  006160  000002                  ROL     2(R0)
013742  006110                          ROL     (R0)
013744  004737  014040                  JSR     PC,GRESET               ;RESET DEFAULT PATTERN IN OUTPUT
                                                                        ;BUFFER.
013750  077330                          SOB     R3,G42
013752  004737  014172                  JSR     PC,GSUM                 ;TYPE ERROR SUMMARY.
2386
2387 013756  000137  014352            JMP     GDONE
2388
2389                            ;USE THIS ROUTINE TO INITIALIZE ALL THE DATA BUFFERS.
2390 013762  012705  014274    GSETUP: MOV     #GFLAG1,R5
2391 013766  012704  000026            MOV     #26,R4
2392 013772  005025            1$:     CLR     (R5)+
2393 013774  077402                    SOB     R4,1$
2394
2395 013776  012705  014310            MOV     #GPAT10,R5
2396 014002  012704  000010            MOV     #10,R4
2397 014006  005125            2$:     COM     (R5)+
2398 014010  077402                    SOB     R4,2$
2399
2400 014012  020037  014300    GS1:    CMP     R0,GPAT00
2401 014016  001401                    BEQ     3$
2402 014020  000207                    RTS     PC
2403
2404 014022  012705  014340    3$:     MOV     #GDAT00,R5
```

```
2405 014026  012704  000004            MOV     #4,R4
2406 014032  005125                4$:  COM     (R5)+
2407 014034  077402                     SOB     R4,4$
2408 014036  000207                     RTS     PC
2409
2410 014040  012705  014340       GRESET: MOV     #GDAT00,R5
2411 014044  012704  000004            MOV     #4,R4
2412 014050  005025                1$:  CLR     (R5)+
2413 014052  077402                     SOB     R4,1$
2414 014054  000137  014012              JMP     GS1
2415
2416                               ;SEE IF THE DATA WRITTEN MATCHES THE DATA READ.
2417 014060  012705  014340       GCMP:   MOV     #GDAT00,R5
2418 014064  012704  000004            MOV     #4,R4
2419 014070  010002                     MOV     R0,R2
2420 014072  022225                1$:  CMP     (R2)+,(R5)+
2421 014074  001402                     BEQ     2$
2422 014076  000137  014106              JMP     GERR1
2423 014102  077405                2$:  SOB     R4,1$
2424 014104  000207                     RTS     PC
2425
2426                               ;COME HERE TO REPORT AND RECORD ERRORS.
2427 014106  012637  014350       GERR1:  MOV     (SP)+,GADR              ;SAVE THE RETURN ADDRESS.
2428 014112  010037  001240            MOV     R0,1240                 ;COMPUTE 'OR' OF BAD DATA.
2429 014116  012705  014320            MOV     #GAND0,R5
2430 014122  012704  000004            MOV     #4,R4
2431 014126  051065  000010       1$:  BIS     (R0),10(R5)
2432 014132  012002                     MOV     (R0)+,R2
2433 014134  005102                     COM     R2
2434 014136  040225                     BIC     R2,(R5)+
2435 014140  077406                     SOB     R4,1$
2436 014142  013700  001240            MOV     1240,R0
2437 014146  005237  014276            INC     GFLAG2                  ;INCREMENT ERROR COUNT.
2438 014152  010037  001240            MOV     R0,$TMP3
2439 014156  012737  014340  001242    MOV     #GDAT00,$TMP4
2440 014164  104044                3$:  ERROR   +44
2441 014166  000177  000156            JMP     @GADR
2442
2443                               ;SEE IF ANY ERRORS HAVE OCCURRED AND WHETHER OR NOT AN ERROR SUMMARY
2444                               ;SHOULD BE TYPED.
2445 014172  005737  014276       GSUM:   TST     GFLAG2                  ;ANY ERRORS?
2446 014176  001435                     BEQ     3$
2447
2448 014200  032777  020000  164732    BIT     #SW13,@SWR              ;INHIBIT ERROR PRINT OUT?
2449 014206  001404                     BEQ     1$
2450 014210  032777  000200  164722    BIT     #SW7,@SWR               ;PRINT SUMMARY?
2451 014216  001425                     BEQ     3$
2452
2453 014220  013737  014276  001246 1$: MOV     GFLAG2,$TMP6            ;YES PRINT SUMMARY.
2454 014226  012737  014320  001240       MOV     #GAND0,$TMP3
2455 014234  012737  014330  001242       MOV     #GOR0,$TMP4
2456 014242  012637  014350            MOV     (SP)+,GADR              ;SAVE RETURN ADDRESS.
2457 014246  012737  014262  001116       MOV     #2$,$ERRPC
2458 014254  112737  000045  001114       MOVB    #45,$ITEMB
2459 014262  004737  040510         2$:  JSR     PC,ERTYPE
2460 014266  000177  000056            JMP     @GADR
2461 014272  000207                3$:  RTS     PC
```

```
 2462
 2463 014274  000000          GFLAG1: .WORD   0
 2464 014276  000000          GFLAG2: .WORD   0
 2465 014300  000000          GPAT00: .WORD   0
 2466 014302  000000          GPAT01: .WORD   0
 2467 014304  000000          GPAT02: .WORD   0
 2468 014306  000000          GPAT03: .WORD   0
 2469 014310  177777          GPAT10: .WORD   -1
 2470 014312  177777          GPAT11: .WORD   -1
 2471 014314  177777          GPAT12: .WORD   -1
 2472 014316  177777          GPAT13: .WORD   -1
 2473 014320  177777          GAND0:  .WORD   -1
 2474 014322  177777          GAND1:  .WORD   -1
 2475 014324  177777          GAND2:  .WORD   -1
 2476 014326  177777          GAND3:  .WORD   -1
 2477 014330  000000          GOR0:   .WORD   0
 2478 014332  000000          GOR1:   .WORD   0
 2479 014334  000000          GOR2:   .WORD   0
 2480 014336  000000          GOR3:   .WORD   0
 2481 014340  000000          GDAT00: .WORD   0
 2482 014342  000000          GDAT01: .WORD   0
 2483 014344  000000          GDAT02: .WORD   0
 2484 014346  000000          GDAT03: .WORD   0
 2485 014350  000000          GADR:   .WORD   0
 2486 014352                  GDONE:
      014352  104413                  RSETUP                  ;GO INITIALIZE THE FPS AND STACK; AND
                                                              ;SEE IF THE USER HAS EXPRESSED
                                                              ;THE DESIRE TO CHANGE THE SOFTWARE
                                                              ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
                                                              ;THE USER TYPED CONTROL G?).
```

```
                                .SBTTL  TEST # 12 - FPP ACCUMULATORS DUAL ADDRESS TEST
                                ;;*****************************************************************
                                ;*TEST 12 - FPP ACCUMULATORS DUAL ADDRESS TEST
                                ;*
                                ;*THIS TEST PERFORMS A DUAL ADDRESSING TEST ON THE FLOATING ACCUMULATORS.
                                ;*NOTE THAT ACCUMULATOR ZERO IS USED TO ACCESS ALL THE OTHERS.
                                ;*
                                ;;*****************************************************************
014354  000004                  TST12:  SCOPE
2494 014356  012737  014364  001110        MOV     #H1,$LPERR      ;SET UP THE LOOP ON ERROR ADDRESS.
2495
2496 014364  005037  015110    H1:     CLR     HFLAG
2497 014370  012700  015112            MOV     #HA1W,R0                ;INITIALIZE THE LOAD BUFFER DATA.
2498 014374  012701  015232            MOV     #HDAT1,R1
2499 014400  012703  000024            MOV     #24,R3
2500 014404  012120            H2:     MOV     (R1)+,(R0)+
2501 014406  077302                    SOB     R3,H2
2502
2503 014410  004737  015036            JSR     PC,HCLR                 ;CLEAR THE OUTPUT DATA BUFFER.
2504
2505 014414  170011            H3:     SETD
2506                            ;LOAD ACCUMULATOR 1
014416  012700  015112                    MOV     #HA1W,R0
014422  172410                          LDD     (R0),AC0
014424  174001                          STD     AC0,AC1
2507                            ;LOAD ACCUMULATOR 2
014426  012700  015122                    MOV     #HA2W,R0
014432  172410                          LDD     (R0),AC0
014434  174002                          STD     AC0,AC2
2508                            ;LOAD ACCUMULATOR 3
014436  012700  015132                    MOV     #HA3W,R0
014442  172410                          LDD     (R0),AC0
014444  174003                          STD     AC0,AC3
2509                            ;LOAD ACCUMULATOR 4
014446  012700  015142                    MOV     #HA4W,R0
014452  172410                          LDD     (R0),AC0
014454  174004                          STD     AC0,AC4
2510                            ;LOAD ACCUMULATOR 5
014456  012700  015152                    MOV     #HA5W,R0
014462  172410                          LDD     (R0),AC0
014464  174005                          STD     AC0,AC5
2511
2512 014466  004737  014722    H4:     JSR     PC,HSTD                 ;GO READ ALL ACCUMULATORS BACK.
2513
2514 014472  004737  015000            JSR     PC,HCMP                 ;SEE IF DATA IS CORRECT.
2515
2516
                                ;COMPLIMENT EACH WORD OF THE DATA STORED IN ACCUMULATOR 1,
                                ;RELOAD THAT ACCUMULATOR, READ ALL THE ACCUMULATORS BACK AND CHECK
                                ;THE DATA.
014476  012700  015112                    MOV     #HA1W,R0
014502  012702  000004                    MOV     #4,R2
014506  010001                          MOV     R0,R1
014510  005121            H5:     COM     (R1)+
014512  172410                          LDD     (R0),AC0
014514  174001                          STD     AC0,AC1
014516  004737  014722                    JSR     PC,HSTD         ;READ ALL THE ACCUMULATORS BACK.
014522  004737  015000                    JSR     PC,HCMP         ;CHECK THE DATA.
```

```
014526  077210                          SOB     R2,H5
2517                                    ;COMPLIMENT EACH WORD OF THE DATA STORED IN ACCUMULATOR 2,
                                        ;RELOAD THAT ACCUMULATOR, READ ALL THE ACCUMULATORS BACK AND CHECK
                                        ;THE DATA.
014530  012700  015122                  MOV     #HA2W,R0
014534  012702  000004                  MOV     #4,R2
014540  010001                          MOV     R0,R1
014542  005121                  H6:     COM     (R1)+
014544  172410                          LDD     (R0),AC0
014546  174002                          STD     AC0,AC2
014550  004737  014722                  JSR     PC,HSTD         ;READ ALL THE ACCUMULATORS BACK.
014554  004737  015000                  JSR     PC,HCMP         ;CHECK THE DATA.
014560  077210                          SOB     R2,H6
2518                                    ;COMPLIMENT EACH WORD OF THE DATA STORED IN ACCUMULATOR 3,
                                        ;RELOAD THAT ACCUMULATOR, READ ALL THE ACCUMULATORS BACK AND CHECK
                                        ;THE DATA.
014562  012700  015132                  MOV     #HA3W,R0
014566  012702  000004                  MOV     #4,R2
014572  010001                          MOV     R0,R1
014574  005121                  H7:     COM     (R1)+
014576  172410                          LDD     (R0),AC0
014600  174003                          STD     AC0,AC3
014602  004737  014722                  JSR     PC,HSTD         ;READ ALL THE ACCUMULATORS BACK.
014606  004737  015000                  JSR     PC,HCMP         ;CHECK THE DATA.
014612  077210                          SOB     R2,H7
2519                                    ;COMPLIMENT EACH WORD OF THE DATA STORED IN ACCUMULATOR 4,
                                        ;RELOAD THAT ACCUMULATOR, READ ALL THE ACCUMULATORS BACK AND CHECK
                                        ;THE DATA.
014614  012700  015142                  MOV     #HA4W,R0
014620  012702  000004                  MOV     #4,R2
014624  010001                          MOV     R0,R1
014626  005121                  H10:    COM     (R1)+
014630  172410                          LDD     (R0),AC0
014632  174004                          STD     AC0,AC4
014634  004737  014722                  JSR     PC,HSTD         ;READ ALL THE ACCUMULATORS BACK.
014640  004737  015000                  JSR     PC,HCMP         ;CHECK THE DATA.
014644  077210                          SOB     R2,H10
2520                                    ;COMPLIMENT EACH WORD OF THE DATA STORED IN ACCUMULATOR 5,
                                        ;RELOAD THAT ACCUMULATOR, READ ALL THE ACCUMULATORS BACK AND CHECK
                                        ;THE DATA.
014646  012700  015152                  MOV     #HA5W,R0
014652  012702  000004                  MOV     #4,R2
014656  010001                          MOV     R0,R1
014660  005121                  H11:    COM     (R1)+
014662  172410                          LDD     (R0),AC0
014664  174005                          STD     AC0,AC5
014666  004737  014722                  JSR     PC,HSTD         ;READ ALL THE ACCUMULATORS BACK.
014672  004737  015000                  JSR     PC,HCMP         ;CHECK THE DATA.
014676  077210                          SOB     R2,H11
2521
2522 014700  005737  015110             TST     HFLAG
2523 014704  001402                     BEQ     H12
2524 014706  000137  015302             JMP     HDONE
2525
2526 014712  005137  015110     H12:    COM     HFLAG
2527 014716  000137  014414             JMP     H3
2528
```

```
2529                                    ;STORE ALL ACCUMULATORS IN THE OUTPUT BUFFERS.
2530 014722  004737  015036      HSTD:   JSR     PC,HCLR                 ;CLEAR ALL OUTPUT BUFFERS.
2531                                    ;STORE ACCUMULATOR 1
     014726  012704  015162              MOV     #HA1R,R4
     014732  172401                      LDD     AC1,AC0
     014734  174014                      STD     AC0,(R4)
2532                                    ;STORE ACCUMULATOR 2
     014736  012704  015172              MOV     #HA2R,R4
     014742  172402                      LDD     AC2,AC0
     014744  174014                      STD     AC0,(R4)
2533                                    ;STORE ACCUMULATOR 3
     014746  012704  015202              MOV     #HA3R,R4
     014752  172403                      LDD     AC3,AC0
     014754  174014                      STD     AC0,(R4)
2534                                    ;STORE ACCUMULATOR 4
     014756  012704  015212              MOV     #HA4R,R4
     014762  172404                      LDD     AC4,AC0
     014764  174014                      STD     AC0,(R4)
2535                                    ;STORE ACCUMULATOR 5
     014766  012704  015222              MOV     #HA5R,R4
     014772  172405                      LDD     AC5,AC0
     014774  174014                      STD     AC0,(R4)
2536 014776  000207                      RTS     PC
2537
2538                                    ;COMPARE DATA LOADED WITH DATA READ.
2539 015000  012637  015106      HCMP:   MOV     (SP)+,HADR              ;SAVE RETURN ADDRESS.
2540 015004  012703  015112              MOV     #HA1W,R3
2541 015010  012704  015162              MOV     #HA1R,R4
2542 015014  012705  000024              MOV     #24,R5
2543 015020  022324              HCMP1:  CMP     (R3)+,(R4)+
2544 015022  001402                      BEQ     HCMP2
2545 015024  000137  015054              JMP     HERROR
2546 015030  077505              HCMP2:  SOB     R5,HCMP1
2547 015032  000177  000050              JMP     @HADR
2548
2549                                    ;CLEAR THE DATA OUTPUT BUFFER.
2550 015036  012704  015162      HCLR:   MOV     #HA1R,R4
2551 015042  012705  000024              MOV     #24,R5
2552 015046  005024              HCLR1:  CLR     (R4)+
2553 015050  077502                      SOB     R5,HCLR1
2554 015052  000207                      RTS     PC
2555
2556                                    ;REPORT ERROR.
2557 015054                      HERROR:
2558 015054  012703  015112              MOV     #HA1W,R3
2559 015060  012704  001236              MOV     #$TMP2,R4
2560 015064  012705  000012              MOV     #12,R5
2561 015070  010324              1$:     MOV     R3,(R4)+
2562 015072  062703  000010              ADD     #10,R3
2563 015076  077504                      SOB     R5,1$
2564 015100  104046              2$:     ERROR   +46
2565 015102  000137  015302              JMP     HDONE
2566
2567
2568 015106  000000              HADR:   .WORD   0
2569 015110  000000              HFLAG:  .WORD   0
2570
```

```
 2571 015112  000000  000000  000000  HA1W:   .WORD   0,0,0,0
 2572 015122  000000  000000  000000  HA2W:   .WORD   0,0,0,0
 2573 015132  000000  000000  000000  HA3W:   .WORD   0,0,0,0
 2574 015142  000000  000000  000000  HA4W:   .WORD   0,0,0,0
 2575 015152  000000  000000  000000  HA5W:   .WORD   0,0,0,0
 2576
 2577 015162  000000  000000  000000  HA1R:   .WORD   0,0,0,0
 2578 015172  000000  000000  000000  HA2R:   .WORD   0,0,0,0
 2579 015202  000000  000000  000000  HA3R:   .WORD   0,0,0,0
 2580 015212  000000  000000  000000  HA4R:   .WORD   0,0,0,0
 2581 015222  000000  000000  000000  HA5R:   .WORD   0,0,0,0
 2582
 2583 015232  073567  073567  073567  HDAT1:  .WORD   73567,73567,73567,73567
 2584 015242  063146  063146  063146  HDAT2:  .WORD   63146,63146,63146,63146
 2585 015252  010421  010421  010421  HDAT3:  .WORD   10421,10421,10421,10421
 2586 015262  031463  031463  031463  HDAT4:  .WORD   31463,31463,31463,31463
 2587 015272  042104  042104  042104  HDAT5:  .WORD   42104,42104,42104,42104
 2588 015302                          HDONE:
      015302  104413                  RSETUP                  ;GO INITIALIZE THE FPS AND STACK; AND
                                                              ;SEE IF THE USER HAS EXPRESSED
                                                              ;THE DESIRE TO CHANGE THE SOFTWARE
                                                              ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
                                                              ;THE USER TYPED CONTROL G?).

 2589
 2590
```

```
                                .SBTTL  TEST # 13 - FSRC MODE 0 WITH ILLEGAL ACCUMULATOR TEST
                                ;;*****************************************************************
                                ;*TEST 13 - FSRC MODE 0 WITH ILLEGAL ACCUMULATOR TEST
                                ;*
                                ;*THIS IS A TEST OF FSRC MODE 0 WITH ACCUMULATORS 6 AND 7. USE OF
                                ;*EITHER OF THESE NON-EXISTENT ACCUMULATORS SHOULD RESULT IN A TRAP TO 244
                                ;*WITH FEC=2 (ILLEGAL FPP INSTRUCTION).
                                ;*
                                ;;*****************************************************************
 015304  000004                 TST13:  SCOPE
2599 015306
 015306  012737  015314  001110         MOV     #1$,$LPERR      ;SET UP THE LOOP ON ERROR ADDRESS.
2600 015314  170011             1$:     SETD                    ;SET FD
2601 015316  012700  016026             MOV     #SPAT10,R0      ;LOAD AC0
2602 015322  172410                     LDD     (R0),AC0
2603
2604 015324  012737  015526  000244     MOV     #SERR0,FPVECT   ;USE OF THE NON-EXISTENT AC-
2605                                                             ;CUMULATOR SHOULD RESULT IN
2606                                                             ;A TRAP TO 244.
2607 015332  012700  000001             MOV     #1,R0           ;A FAILURE IN THE FSRC FLOWS
2608                                                             ;WILL RESULT IN AN ODD ADDRESS
2609 015336  012737  015736  000004     MOV     #SERR1,ERRVECT  ;TRAP TO 4.
2610 015344  005003                     CLR     R3
2611
2612 015346  172407             S2:     LDD     AC7,AC0
2613 015350  170000             S3:     CFCC
2614 015352  005203                     INC     R3
2615 015354  005203             S4:     INC     R3
2616
2617 015356  012701  016036             MOV     #SDAT00,R1      ;NO TRAP OCCURRED!!
2618 015362  174011                     STD     AC0,(R1)        ;SEE IF AC0 WAS MODIFIED.
2619
2620 015364  012701  016036             MOV     #SDAT00,R1
2621 015370  012702  016026             MOV     #SPAT10,R2
2622 015374  012703  000004             MOV     #4,R3
2623 015400  022122             S5:     CMP     (R1)+,(R2)+
2624 015402  001402                     BEQ     S6
2625 015404  000137  015666             JMP     SERR2
2626 015410  077305             S6:     SOB     R3,S5
2627
2628 015412  000137  015712             JMP     SERR3
2629
2630                            ;NOW TEST AC6.
2631 015416                     S7:
 015416  012737  015424  001110         MOV     #1$,$LPERR      ;SET UP THE LOOP ON ERROR ADDRESS.
2632 015424  170011             1$:     SETD
2633
2634 015426  012700  016026             MOV     #SPAT10,R0      ;LOAD AC0
2635 015432  172410                     LDD     (R0),AC0
2636
2637 015434  012737  015604  000244     MOV     #SERR4,FPVECT
2638 015442  012700  000001             MOV     #1,R0
2639 015446  012737  015770  000004     MOV     #SERR5,ERRVECT
2640 015454  005003                     CLR     R3
2641
2642 015456  172406             S8:     LDD     AC6,AC0
2643 015460  170000             S9:     CFCC
```

```
2644 015462  005203                        INC     R3
2645 015464  005203                S10:    INC     R3
2646
2647 015466  012701  016036                MOV     #SDAT00,R1
2648 015472  174011                        STD     AC0,(R1)                ;NO TRAP! GET AC0.
2649
2650 015474  012701  016036                MOV     #SDAT00,R1              ;WAS AC0 MODIFIED.
2651 015500  012702  016026                MOV     #SPAT10,R2
2652 015504  012703  000004                MOV     #4,R3
2653 015510  022122                S11:    CMP     (R1)+,(R2)+
2654 015512  001402                        BEQ     S12
2655 015514  000137  015700                JMP     SERR6
2656 015520  077305                S12:    SOB     R3,S11
2657 015522  000137  015724                JMP     SERR7
2658
2659                               ;TRAPPED TO 244.
2660 015526  021627  015350        SERR0:  CMP     (SP),#S3                ;PC OF TRAP CORRECT?
2661 015532  001402                        BEQ     1$
2662 015534  000137  041142                JMP     FPSPUR
2663
2664 015540  012737  015416  016022  1$:   MOV     #S7,SADR
2665
2666 015546  011637  001236        SERR10: MOV     (SP),$TMP2
2667 015552  022626                        CMP     (SP)+,(SP)+
2668 015554  005004                        CLR     R4
2669 015556  170204                        STFPS   R4                      ;IS FPS CORRECT?
2670 015560  022704  100200                CMP     #100200,R4
2671 015564  001020                        BNE     SERR15
2672
2673 015566  005004                        CLR     R4
2674 015570  170304                        STST    R4                      ;IS FEC CORRECT?
2675 015572  022704  000002                CMP     #2,R4
2676 015576  001023                        BNE     SERR20
2677 015600  000177  000216                JMP     @SADR
2678
2679 015604  021627  015460        SERR4:  CMP     (SP),#S9
2680 015610  001402                        BEQ     1$
2681 015612  000137  041142                JMP     FPSPUR
2682 015616  012737  016046  016022  1$:   MOV     #SDONE,SADR
2683 015624  000750                        BR      SERR10
2684
2685                               ;REPORT FPS FAILURE:
2686 015626  012737  100200  001242  SERR15: MOV     #100200,$TMP4
2687 015634  010437  001240                MOV     R4,$TMP3
2688 015640  104117                1$:     ERROR   +117
2689 015642  000177  000154                JMP     @SADR
2690
2691                               ;REPORT FEC BAD:
2692 015646  012737  000002  001242  SERR20: MOV     #2,$TMP4
2693 015654  010437  001240                MOV     R4,$TMP3
2694 015660  104120                1$:     ERROR   +120
2695 015662  000177  000134                JMP     @SADR
2696
2697
2698                               ;AC0 WAS MODIFIED. (BUT FSRC) FORK FAILED.
2699 015666  012737  015346  001236  SERR2:  MOV     #S2,$TMP2
2700 015674  104112                1$:     ERROR   +112
```

```
2701 015676  000463                          BR      SDONE
2702 015700  012737  015456  001236  SERR6:  MOV     #S8,$TMP2
2703 015706  104114                  1$:     ERROR   +114
2704 015710  000456                          BR      SDONE
2705
2706 015712  012737  015346  001236  SERR3:  MOV     #S2,$TMP2
2707 015720  104111                  1$:     ERROR   +111
2708 015722  000451                          BR      SDONE
2709 015724  012737  015456  001236  SERR7:  MOV     #S8,$TMP2
2710 015732  104113                  1$:     ERROR   +113
2711 015734  000444                          BR      SDONE
2712
2713                                 ;FAILURE OF (BUT FSRC) CAUSED AN ODD ADDRESS TRAP TO 4.
2714 015736  021627  015350          SERR1:  CMP     (SP),#S3                ;DID TRAP OCCUR ON TESTED INSTRUCTION?
2715 015742  001405                          BEQ     1$
2716 015744  021627  015354                  CMP     (SP),#S4
2717 015750  001402                          BEQ     1$
2718 015752  000137  041174                  JMP     CPSPUR
2719
2720 015756  011637  001236          1$:     MOV     (SP),$TMP2
2721 015762  022626                          CMP     (SP)+,(SP)+
2722 015764  104115                  2$:     ERROR   +115
2723 015766  000427                          BR      SDONE
2724
2725 015770  021627  015456          SERR5:  CMP     (SP),#S8                ;DID TRAP OCCUR ON TEST INSTRUCTION?
2726 015774  001405                          BEQ     1$
2727 015776  021627  015460                  CMP     (SP),#S9
2728 016002  001402                          BEQ     1$
2729 016004  000137  041174                  JMP     CPSPUR
2730
2731 016010  011637  001236          1$:     MOV     (SP),$TMP2
2732 016014  022626                          CMP     (SP)+,(SP)+
2733 016016  104116                  2$:     ERROR   +116
2734 016020  000412                          BR      SDONE
2735
2736 016022  000000                  SADR:   0
2737 016024  177777                          -1
2738 016026  010421                  SPAT10: .WORD   10421
2739 016030  021042                  SPAT11: .WORD   21042
2740 016032  031463                  SPAT12: .WORD   31463
2741 016034  042104                  SPAT13: .WORD   42104
2742
2743 016036  000000                  SDAT00: .WORD   0
2744 016040  000000                  SDAT01: .WORD   0
2745 016042  000000                  SDAT02: .WORD   0
2746 016044  000000                  SDAT03: .WORD   0
2747
2748 016046                          SDONE:
     016046  104413                          RSETUP                  ;GO INITIALIZE THE FPS AND STACK; AND
                                                                     ;SEE IF THE USER HAS EXPRESSED
                                                                     ;THE DESIRE TO CHANGE THE SOFTWARE
                                                                     ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
                                                                     ;THE USER TYPED CONTROL G?).
2749
```

```
                                        .SBTTL  TEST # 14 - FSRC MODE 2 TEST
                                        ;;*********************************************************************
                                        ;*TEST 14 - FSRC MODE 2 TEST
                                        ;*
                                        ;* THIS IS A TEST OF FSRC MODE 2, AUTO
                                        ;* INCREMENT MODE.
                                        ;*
                                        ;;*********************************************************************
 016050  000004                         TST14:  SCOPE
2757 016052  012737  016060  001110            MOV     #J1,$LPERR      ;SET UP THE LOOP ON ERROR ADDRESS.
2758
2759 016060                             J1:
2760 016060  170011                            SETD                    ;SET DOUBLE MODE
2761
2762 016062  012700  016336                    MOV     #JDAT0,R0
2763 016066  172410                            LDD     (R0),AC0        ;LOAD AC0
2764
2765 016070  012700  016316                    MOV     #JDATI0,R0
2766 016074  005003                            CLR     R3
2767 016076  012737  016166  000004            MOV     #JERR0,ERRVECT
2768
2769 016104  172420                     J2:    LDD     (R0)+,AC0       ;TEST INSTRUCTION
2770 016106  005203                     J3:    INC     R3
2771 016110  005203                     J4:    INC     R3
2772
2773 016112  012701  016326                    MOV     #JDAT00,R1
2774 016116  174011                            STD     AC0,(R1)        ;PICK UP RESULTS
2775
2776 016120  020027  016306                    CMP     R0,#JBUF0       ;WAS AN AUTO
2777 016124  001001                            BNE     1$              ;DECREMENT EXECUTED?
2778 016126  000442                            BR      JERR1
2779
2780 016130  012702  016316             1$:    MOV     #JDATI0,R2      ;IS DATA CORRECT?
2781 016134  012703  016326                    MOV     #JDAT00,R3
2782 016140  012704  000004                    MOV     #4,R4
2783 016144  022223                     J5:    CMP     (R2)+,(R3)+
2784 016146  001401                            BEQ     J6
2785 016150  000443                            BR      JERR2
2786 016152  077404                     J6:    SOB     R4,J5
2787
2788 016154  022700  016326                    CMP     #JDATI0+10,R0   ;WAS R0 INCREM.
2789 016160  001401                            BEQ     J7              ;BY 10 (OCTAL)
2790 016162  000424                            BR      JERR1
2791
2792 016164  000470                     J7:    BR      JDONE
2793
2794                                     ;IF A TRAP THROUGH 4 OCCURS COME HERE
2795
2796 016166  021627  016106             JERR0: CMP     (SP),#J3        ;SEE IF THE TRAP
2797 016172  001405                            BEQ     J10             ;OCCURRED ON THE
2798 016174  021627  016110                    CMP     (SP),#J4        ;TESTED INSTRUCTION
2799 016200  001402                            BEQ     J10
2800 016202  000137  041174                    JMP     CPSPUR
2801
2802 016206  012737  000762  001240     J10:   MOV     #762,$TMP3      ;REPORT FSRC FLOW
2803 016214  012737  000322  001242            MOV     #322,$TMP4      ;FAILURE
2804 016222  011637  001236                    MOV     (SP),$TMP2
```

```
2805 016226  022626                        CMP     (SP)+,(SP)+
2806 016230  104052                1$:     ERROR   +52
2807 016232  000445                        BR      JDONE
2808
2809 016234                        JERR1:                          ;REPORT, R0 NOT
2810 016234  012737  016104  001236        MOV     #J2,$TMP2       ;CORRECTLY AFFECTED
2811 016242  010037  001240                MOV     R0,$TMP3
2812 016246  012737  016326  001242        MOV     #JDATI0+10,$TMP4
2813 016254  104053                1$:     ERROR   +53
2814 016256  000433                        BR      JDONE
2815
2816                               ;REPORT DATA FAILURE
2817
2818 016260                        JERR2:
2819 016260  012737  016104  001236        MOV     #J2,$TMP2
2820 016266  012737  016316  001240        MOV     #JDATI0,$TMP3
2821 016274  012737  016326  001242        MOV     #JDAT00,$TMP4
2822 016302  104054                1$:     ERROR   +54
2823 016304  000420                        BR      JDONE
2824
2825 016306  010421                JBUF0:  .WORD   010421
2826 016310  021042                JBUF1:  .WORD   021042
2827 016312  042104                JBUF2:  .WORD   042104
2828 016314  031463                JBUF3:  .WORD   031463
2829
2830 016316  052525                JDATI0: .WORD   052525
2831 016320  114631                JDATI1: .WORD   114631
2832 016322  063146                JDATI2: .WORD   063146
2833 016324  073567                JDATI3: .WORD   073567
2834
2835 016326  000000                JDAT00: .WORD   0
2836 016330  000000                JDAT01: .WORD   0
2837 016332  000000                JDAT02: .WORD   0
2838 016334  000000                JDAT03: .WORD   0
2839
2840 016336  177777                JDAT0:  .WORD   -1
2841 016340  177777                JDAT1:  .WORD   -1
2842 016342  177777                JDAT2:  .WORD   -1
2843 016344  177777                JDAT3:  .WORD   -1
2844
2845
2846 016346                        JDONE:
     016346  104413                        RSETUP                  ;GO INITIALIZE THE FPS AND STACK; AND
                                                                    ;SEE IF THE USER HAS EXPRESSED
                                                                    ;THE DESIRE TO CHANGE THE SOFTWARE
                                                                    ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
                                                                    ;THE USER TYPED CONTROL G?).

2847
2854
```

```
                                        .SBTTL  TEST # 15 - FSRC MODE 4 TEST
                                        ;;*****************************************************************
                                        ;*TEST 15 - FSRC MODE 4 TEST
                                        ;*
                                        ;* THIS IS A TEST OF FSRC MODE 4, AUTO
                                        ;* DECREMENT MODE.
                                        ;*
                                        ;;*****************************************************************
016350  000004                          TST15:  SCOPE
2856 016352  012737  016360  001110            MOV     #K1,$LPERR      ;SET UP THE LOOP ON ERROR ADDRESS.
2857
2858 016360                             K1:
2859 016360  170011                            SETD                    ;SET DOUBLE MODE
2860
2861 016362  012700  016634                    MOV     #KPAT0,R0
2862 016366  172410                            LDD     (R0),AC0        ;LOAD A DEFAULT
2863                                                                    ;PATTERN INTO AC0
2864 016370  012700  016614                    MOV     #KBUF0,R0
2865 016374  005003                            CLR     R3
2866 016376  012737  016466  000004            MOV     #KERR0,ERRVECT
2867
2868 016404  172440                     K2:    LDD     -(R0),AC0       ;TEST INSTRUCTION
2869 016406  005203                     K3:    INC     R3
2870 016410  005203                     K4:    INC     R3
2871
2872 016412  012701  016624                    MOV     #KDAT00,R1
2873 016416  174011                            STD     AC0,(R1)        ;PICK UP THE RESULT
2874
2875 016420  020027  016624                    CMP     R0,#KBUF0+10    ;WAS AN AUTO
2876 016424  001001                            BNE     1$              ;INCREMENT EXECUTED
2877 016426  000441                            BR      KERR1
2878
2879 016430  012702  016604             1$:    MOV     #KDAT10,R2      ;IS DATA CORRECT?
2880 016434  012703  016624                    MOV     #KDAT00,R3
2881 016440  012704  000004                    MOV     #4,R4
2882 016444  022223                     K5:    CMP     (R2)+,(R3)+
2883 016446  001401                            BEQ     K6
2884 016450  000442                            BR      KERR2
2885 016452  077404                     K6:    SOB     R4,K5
2886
2887 016454  022700  016604                    CMP     #KBUF0-10,R0    ;WAS R0 DECREMENTED
2888 016460  001401                            BEQ     K7              ;PROPERLY?
2889 016462  000423                            BR      KERR1
2890
2891 016464  000467                     K7:    BR      KDONE
2892
2893                                     ;TRAP TO HERE ON AN ODD ADDRESS ERROR
2894
2895 016466  021627  016406             KERR0: CMP     (SP),#K3        ;SEE IF THE ERROR
2896 016472  001405                            BEQ     K10             ;OCCURRED AT THE
2897 016474  021627  016410                    CMP     (SP),#K4        ;INSTRUCTION TESTED.
2898 016500  001402                            BEQ     K10
2899 016502  000137  041174                    JMP     CPSPUR
2900
2901 016506  012737  000762  001240     K10:   MOV     #762,$TMP3      ;REPORT FAILURE IN
2902 016514  012737  000324  001242            MOV     #324,$TMP4      ;FSRC FLOWS
2903 016522  011637  001236                    MOV     (SP),$TMP2
```

```
2904 016526  104055                1$:     ERROR   +55
2905 016530  000445                        BR      KDONE
2906
2907 016532                        KERR1:                          ;REPORT, R0
2908 016532  012737  016404  001236        MOV     #K2,$TMP2       ;INCORRECTLY AFFECTED.
2909 016540  010037  001240                MOV     R0,$TMP3
2910 016544  012737  016604  001242        MOV     #KDATI0,$TMP4
2911 016552  104056                1$:     ERROR   +56
2912 016554  000433                        BR      KDONE
2913
2914                               ;REPORT DATA FAILURE
2915
2916 016556                        KERR2:
2917 016556  012737  016404  001236        MOV     #K2,$TMP2
2918 016564  012737  016604  001240        MOV     #KDATI0,$TMP3
2919 016572  012737  016624  001242        MOV     #KDATO0,$TMP4
2920 016600  104057                1$:     ERROR   +57
2921 016602  000420                        BR      KDONE
2922
2923 016604  052525                KDATI0: .WORD   052525
2924 016606  114631                KDATI1: .WORD   114631
2925 016610  063140                KDATI2: .WORD   063140
2926 016612  073567                KDATI3: .WORD   073567
2927
2928 016614  010421                KBUF0:  .WORD   010421
2929 016616  031463                KBUF1:  .WORD   031463
2930 016620  042104                KBUF2:  .WORD   042104
2931 016622  021042                KBUF3:  .WORD   021042
2932
2933 016624  000000                KDATO0: .WORD   0
2934 016626  000000                KDATO1: .WORD   0
2935 016630  000000                KDATO2: .WORD   0
2936 016632  000000                KDATO3: .WORD   0
2937
2938 016634  177777                KPATO:  .WORD   -1
2939 016636  177777                KPAT1:  .WORD   -1
2940 016640  177777                KPAT2:  .WORD   -1
2941 016642  177777                DPAT3:  .WORD   -1
2942
2943 016644                        KDONE:
     016644  104413                        RSETUP                  ;GO INITIALIZE THE FPS AND STACK; AND
                                                                    ;SEE IF THE USER HAS EXPRESSED
                                                                    ;THE DESIRE TO CHANGE THE SOFTWARE
                                                                    ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
                                                                    ;THE USER TYPED CONTROL G?).
2944
2951
```

```
                                .SBTTL  TEST # 16 - FSRC MODE 2, WITH FD=0, TEST
                                ;;*****************************************************************
                                ;*TEST 16 - FSRC MODE 2, WITH FD=0, TEST
                                ;*
                                ;* THIS IS A TEST OF FSRC MODE 2 WITH
                                ;* FD=0.  (AUTO INCREMENT)
                                ;*
                                ;;*****************************************************************
     016646  000004             TST16:  SCOPE
2953 016650  012737  016656  001110        MOV     #L1,$LPERR      ;SET UP THE LOOP ON ERROR ADDRESS.
2954
2955 016656                     L1:
2956 016656  170011                        SETD                    ;SET DOUBLE MODE
2957
2958 016660  012700  017126                MOV     #LPAT10,R0
2959 016664  172410                        LDD     (R0),AC0        ;LOAD AC0
2960
2961 016666  012700  017150                MOV     #LDAT10,R0      ;SET UP THE INPUT
2962 016672  012701  017136                MOV     #LPAT20,R1      ;DATA
2963 016676  012702  000004                MOV     #4,R2
2964
2965 016702  012120             1$:        MOV     (R1)+,(R0)+
2966 016704  077202                        SOB     R2,1$
2967
2968 016706  012700  017150                MOV     #LDAT10,R0
2969 016712  005003                        CLR     R3
2970 016714  170001                        SETF                    ;CLEAR FD.
2971
2972 016716  172420             L2:        LDF     (R0)+,AC0
2973 016720  005203             L3:        INC     R3
2974
2975 016722                     L4:
2976 016722  170011                        SETD                    ;SET FD
2977
2978 016724  012701  017162                MOV     #LDAT00,R1
2979 016730  174011                        STD     AC0,(R1)        ;PICK UP RESULTS
2980
2981 016732  020027  017154                CMP     R0,#LDAT12      ;WAS R0 INCREMENTED
2982 016736  001401                        BEQ     1$              ;CORRECTLY BY 4
2983 016740  000421                        BR      LERR1
2984
2985 016742  012737  177777  017154  1$:  MOV     #-1,LDAT12
2986 016750  012737  177777  017156       MOV     #-1,LDAT13
2987 016756  012702  017150                MOV     #LDAT10,R2      ;IS DATA CORRECT
2988 016762  012703  017162                MOV     #LDAT00,R3
2989 016766  012704  000004                MOV     #4,R4
2990
2991 016772  022223             L5:        CMP     (R2)+,(R3)+
2992 016774  001401                        BEQ     L6
2993 016776  000427                        BR      LERR2
2994 017000  077404             L6:        SOB     R4,L5
2995
2996 017002  000473                        BR      LDONE
2997
2998 017004                     LERR1:                          ;REPORT FAILURE
2999 017004  012737  016716  001236        MOV     #L2,$TMP2       ;R0 NOT INCREMENTED
3000 017012  010037  001240                MOV     R0,$TMP3        ;BY 4
```

```
3001 017016  012737  017154  001242          MOV     #LDATI2,$TMP4
3002 017024  104060                  1$:     ERROR   +60
3003 017026  000461                          BR      LDONE
3004
3005 017030                          LERR3:                          ;REPORT DATA FAILURE.
3006 017030  012737  016716  001236          MOV     #L2,$TMP2
3007 017036  012737  017150  001240          MOV     #LDATI0,$TMP3
3008 017044  012737  017162  001242          MOV     #LDAT00,$TMP4
3009 017052  104061                  1$:     ERROR   +61
3010 017054  000446                          BR      LDONE
3011
3012 017056  012702  017136          LERR2:  MOV     #LPAT20,R2      ;DID (BUT FD)
3013 017062  012703  017162                  MOV     #LDAT00,R3      ;FAIL.
3014 017066  012704  000004                  MOV     #4,R4
3015 017072  022223                  1$:     CMP     (R2)+,(R3)+
3016 017074  001355                          BNE     LERR3
3017 017076  077403                          SOB     R4,1$
3018 017100  012737  016716  001236          MOV     #L2,$TMP2
3019 017106  012737  017150  001240          MOV     #LDATI0,$TMP3
3020 017114  012737  017164  001242          MOV     #LDAT01,$TMP4
3021 017122  104062                  2$:     ERROR   +62
3022 017124  000422                          BR      LDONE
3023
3024 017126  177777                  LPAT10: .WORD   -1
3025 017130  177777                  LPAT11: .WORD   -1
3026 017132  177777                  LPAT12: .WORD   -1
3027 017134  177777                  LPAT13: .WORD   -1
3028
3029 017136  052525                  LPAT20: .WORD   052525
3030 017140  114631                  LPAT21: .WORD   114631
3031 017142  063142                  LPAT22: .WORD   063142
3032 017144  073567  000001          LPAT23: .WORD   073567,1
3033 017150  000000                  LDATI0: .WORD   0
3034 017152  000000                  LDATI1: .WORD   0
3035 017154  000000                  LDATI2: .WORD   0
3036 017156  000000  000001          LDATI3: .WORD   0,1
3037 017162  000000                  LDAT00: .WORD   0
3038 017164  000000                  LDAT01: .WORD   0
3039 017166  000000                  LDAT02: .WORD   0
3040 017170  000000                  LDAT03: .WORD   0
3041
3042 017172                          LDONE:
     017172  104413                          RSETUP                  ;GO INITIALIZE THE FPS AND STACK; AND
                                                                     ;SEE IF THE USER HAS EXPRESSED
                                                                     ;THE DESIRE TO CHANGE THE SOFTWARE
                                                                     ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
                                                                     ;THE USER TYPED CONTROL G?).
3043
3051
```

TEST # 17 - FSRC MODE 2 WITH GR7, IMMEDIATE MODE, TEST

```
3052                                    .SBTTL  TEST # 17 - FSRC MODE 2 WITH GR7, IMMEDIATE MODE, TEST
                                        ;;*****************************************************************
                                        ;*TEST 17 - FSRC MODE 2 WITH GR7, IMMEDIATE MODE, TEST
                                        ;*
                                        ;* THIS IS A TEST OF FSRC MODE 2
                                        ;* USING GR7 (THE PC).  THIS IS IMMEDIATE
                                        ;* MODE.
                                        ;*
                                        ;;*****************************************************************
     017174  000004                     TST17:  SCOPE
3053
3054 017176                             M1:
3055 017176  170011                             SETD
3056
3057 017200  012700  017472                     MOV     #MPAT10,R0
3058 017204  172410                             LDD     (R0),AC0        ;LOAD BACKROUND
3059                                                                     ;PATTERN INTO AC0.
3060 017206  005004                             CLR     R4
3061 017210  012737  017432  000004             MOV     #MERR3,ERRVECT
3062
3063 017216  172427  000000             M15:    LDD     #0,AC0          ;TEST INSTRUCTION
3064         017220                             .=.-2
3065 017220  005204                             .WORD   5204
3066 017222  005204                     M2:     INC     R4              ;NOTE THAT
3067 017224  005204                     M3:     INC     R4              ;005204=INC R4
3068 017226  005204                     M4:     INC     R4
3069
3070 017230  020427  000003                     CMP     R4,#3           ;SEE IF THE PC
3071 017234  001401                             BEQ     1$              ;WAS INCREMENTED
3072 017236  000443                             BR      MERR0           ;BY 2 DURING THE
3073                                                                     ;INSTRUCTION.  IF
3074                                                                     ;NOT THEN A BAD
3075                                                                     ;CONSTANT WAS GENERATED
3076 017240  012700  017512             1$:     MOV     #MDAT00,R0
3077 017244  174010                             STD     AC0,(R0)        ;GET THE DATA
3078
3079 017246  012700  017512                     MOV     #MDAT00,R0
3080 017252  022720  005204                     CMP     #5204,(R0)+     ;IS THE DATA CORRECT?
3081 017256  001401                             BEQ     M5
3082 017260  000451                             BR      MERR1
3083 017262  012701  000003             M5:     MOV     #3,R1
3084 017266  005720                     M6:     TST     (R0)+
3085 017270  001002                             BNE     M7
3086 017272  077103                             SOB     R1,M6
3087 017274  000512                             BR      MDONE
3088
3089 017276  012700  017512             M7:     MOV     #MDAT00,R0      ;DID (BUT GRM) FAIL?
3090 017302  012701  000004                     MOV     #4,R1
3091 017306  022720  005204             M8:     CMP     #5204,(R0)+
3092 017312  001401                             BEQ     M9
3093 017314  000433                             BR      MERR1
3094 017316  077105                     M9:     SOB     R1,M8
3095
3096 017320                             MERR2:                          ;REPORT FAILURE
3097 017320  012737  017216  001236             MOV     #M15,$TMP2      ;OF (BUT GR7)
3098 017326  012737  017502  001240             MOV     #MPAT20,$TMP3
3099 017334  012737  017512  001242             MOV     #MDAT00,$TMP4
```

```
3100 017342  104063                1$:     ERROR   +63
3101 017344  000466                        BR      MDONE
3102
3103 017346  012705  017222        MERR0:  MOV     #M2,R5          ;REPORT FAILURE
3104 017352  010537  001242                MOV     R5,$TMP4        ;PC INCREMENTED
3105 017356  162704  000003                SUB     #3,R4
3106 017362  006304                        ASL     R4
3107 017364  160405                        SUB     R4,R5
3108 017366  010537  001240                MOV     R5,$TMP3
3109 017372  012737  017216  001236        MOV     #M15,$TMP2
3110 017400  104064                1$:     ERROR   +64
3111 017402  000447                        BR      MDONE
3112
3113 017404                        MERR1:                          ;REPORT DATA
3114 017404  012737  017216  001236        MOV     #M15,$TMP2      ;FAILURE
3115 017412  012737  017512  001240        MOV     #MDAT00,$TMP3
3116 017420  012737  017502  001242        MOV     #MPAT20,$TMP4
3117 017426  104066                1$:     ERROR   +66
3118 017430  000434                        BR      MDONE
3119                               ;TRAP TO HERE THROUGH 4.
3120 017432  032716  000001        MERR3:  BIT     #1,(SP)         ;SEE IF THE
3121 017436  001002                        BNE     1$              ;TRAP TO 4 OCCURRED
3122 017440  000137  041174                JMP     CPSPUR          ;BECAUSE OF AN
3123                                                               ;ODD ADDRESS
3124 017444  011637  001240        1$:     MOV     (SP),$TMP3      ;IF YES REPORT
3125 017450  012737  017222  001242        MOV     #M2,$TMP4       ;BAD CONSTANT
3126 017456  012737  017216  001236        MOV     #M15,$TMP2      ;GENERATED
3127 017464  022626                        CMP     (SP)+,(SP)+
3128 017466  104065                2$:     ERROR   +65
3129 017470  000414                        BR      MDONE
3130
3131 017472  177777        MPAT10: .WORD   -1
3132 017474  177777        MPAT11: .WORD   -1
3133 017476  177777        MPAT12: .WORD   -1
3134 017500  177777        MPAT13: .WORD   -1
3135
3136 017502  005204        MPAT20: .WORD   5204
3137 017504  005204        MPAT21: .WORD   5204
3138 017506  005204        MPAT22: .WORD   5204
3139 017510  005204        MPAT23: .WORD   5204
3140
3141 017512  000000        MDAT00: .WORD   0
3142 017514  000000        MDAT01: .WORD   0
3143 017516  000000        MDAT02: .WORD   0
3144 017520  000000        MDAT03: .WORD   0
3145
3146 017522                MDONE:
     017522  104413                RSETUP                  ;GO INITIALIZE THE FPS AND STACK; AND
                                                            ;SEE IF THE USER HAS EXPRESSED
                                                            ;THE DESIRE TO CHANGE THE SOFTWARE
                                                            ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
                                                            ;THE USER TYPED CONTROL G?).
3147
3154
```

```
                                .SBTTL  TEST # 20 - FSRC MODE 3 TEST
                                ;;*****************************************************************
                                ;*TEST 20 - FSRC MODE 3 TEST
                                ;*
                                ;* THIS IS A TEST OF FSRC MODE 3, AUTO INCREMENT
                                ;* DEFERRED
                                ;*
                                ;;*****************************************************************
 017524  000004                 TST20:  SCOPE

 017526                         N1:
 017526  170011                         SETD                    ;SET FD MODE

 017530  012700  020210                 MOV     #NPAT10,R0
 017534  172410                 LDD     (R0),AC0                ;LOAD AC0 WITH A DEFAULT
                                                                ;PATTERN
 017536  012700  020176                 MOV     #NPAT20,R0
 017542  005003                 CLR     R3
 017544  012737  017720  000004 MOV     #NERR0,ERRVECT          ;IF A FAILURE OCCURS
                                                                ;IN THE FSRC FLOWS AN
                                                                ;ODD TRAP TO 4 COULD OCCUR
 017552  172430                 N2:     LDD     @(R0)+,AC0      ;TEST INSTRUCTION.
 017554  005203                 N3:     INC     R3
 017556  005203                 N4:     INC     R3

 017560  012701  020156                 MOV     #NDAT00,R1
 017564  174011                 STD     AC0,(R1)                ;GET THE DATA

 017566  020027  020200                 CMP     R0,#NPAT20+2    ;WAS R0 INCREMENTED
 017572  001437                 BEQ     N12                     ;BY 2?

 017574  020027  020206         N5:     CMP     R0,#NPAT20+10   ;FSRC MODE 2?
 017600  001001                 BNE     N6
 017602  000506                 BR      NERR1

 017604  020027  020166         N6:     CMP     R0,#NPAT20-10   ;FSRC MODE 4?
 017610  001001                 BNE     N7
 017612  000520                 BR      NERR2

 017614  020027  020176         N7:     CMP     R0,#NPAT20
 017620  001023                 BNE     N11

 017622  012702  020156                 MOV     #NDAT00,R2      ;FSRC MODE 0?
 017626  012703  000004                 MOV     #4,R3
 017632  022227  177777         N8:     CMP     (R2)+,#-1
 017636  001002                 BNE     N9
 017640  077304                 SOB     R3,N8
 017642  000510                 BR      NERR3

 017644  012702  020156         N9:     MOV     #NDAT00,R2      ;FSRC MODE 1
 017650  012703  020176                 MOV     #NPAT20,R3
 017654  012704  000004                 MOV     #4,R4
 017660  022223                 N10:    CMP     (R2)+,(R3)+
 017662  001002                 BNE     N11
 017664  077403                 SOB     R4,N10
 017666  000502                 BR      NERR4
```

```
3204 017670  000505                  N11:    BR      NERR5
3205
3206 017672  012702  020156          N12:    MOV     #NDAT00,R2      ;DATA CORRECT?
3207 017676  012703  020220                  MOV     #NDAT10,R3
3208 017702  012704  000004                  MOV     #4,R4
3209 017706  022223                  N13:    CMP     (R2)+,(R3)+
3210 017710  001002                          BNE     N14
3211 017712  077403                          SOB     R4,N13
3212 017714  000545                          BR      NDONE
3213
3214 017716  000504                  N14:    BR      NERR6
3215
3216                                 ;IF AN ODD ADDRESS TRAP OCCURS COME HERE
3217                                 ;TO SEE IF THE FAILURE WAS IN THE FSRC
3218                                 ;FLOWS
3219
3220 017720  022716  017556          NERR0:  CMP     #N4,(SP)        ;FSRC MODE 6 OR 7?
3221 017724  001412                          BEQ     NERR10
3222 017726  022716  017554                  CMP     #N3,(SP)
3223 017732  001402                          BEQ     1$
3224 017734  000137  041174                  JMP     CPSPUR
3225 017740  020027  020174          1$:     CMP     R0,#NPAT20-2    ;FSRC MODE 5?
3226 017744  001407                          BEQ     NERR11
3227 017746  000137  041174                  JMP     CPSPUR
3228
3229 017752                          NERR10:                         ;WENT TO FSRC
3230 017752  011637  001236                  MOV     (SP),$TMP2      ;MODE 6 OR 7.
3231 017756  022626                          CMP     (SP)+,(SP)+
3232 017760  104067                  1$:     ERROR   +67
3233 017762  000522                          BR      NDONE
3234
3235 017764  011637  001236          NERR11: MOV     (SP),$TMP2      ;WENT TO FSRC
3236 017770  022626                          CMP     (SP)+,(SP)+     ;MODE 5.
3237 017772  012737  000627  001244          MOV     #627,$TMP5
3238 020000  012737  000323  001250          MOV     #323,$TMP7
3239 020006  012737  000325  001246          MOV     #325,$TMP6
3240 020014  104070                  1$:     ERROR   +70
3241 020016  000504                          BR      NDONE
3242 020020  012737  000322  001246  NERR1:  MOV     #322,$TMP6      ;FSRC MODE 2.
3243 020026  012737  000627  001244  NERR20: MOV     #627,$TMP5
3244 020034  012737  000323  001250          MOV     #323,$TMP7
3245 020042  012737  017552  001236          MOV     #N2,$TMP2
3246 020050  104071                  1$:     ERROR   +71
3247 020052  000466                          BR      NDONE
3248 020054  012737  000324  001246  NERR2:  MOV     #324,$TMP6      ;FSRC MODE 4
3249 020062  000761                          BR      NERR20
3250 020064  012737  000320  001246  NERR3:  MOV     #320,$TMP6      ;FSRC MODE 0
3251 020072  000755                          BR      NERR20
3252 020074  012737  000321  001246  NERR4:  MOV     #321,$TMP6      ;FSRC MODE 1
3253 020102  000751                          BR      NERR20
3254
3255 020104  010037  001240          NERR5:  MOV     R0,$TMP3        ;R0 NOT
3256 020110  012737  020200  001242          MOV     #NPAT20+2,$TMP4 ;INCREMENTED
3257 020116  012737  017552  001236          MOV     #N2,$TMP2       ;PROPERLY.
3258 020124  104072                  1$:     ERROR   +72
3259 020126  000440                          BR      NDONE
3260
```

```
3261 020130                                NERR6:                                  ;DATA FAILURE.
3262 020130  012737  017552  001236                MOV     #N2,$TMP2
3263 020136  012737  020156  001240                MOV     #NDAT00,$TMP3
3264 020144  012737  020220  001242                MOV     #NDAT10,$TMP4
3265 020152  104073                        1$:     ERROR   +73
3266 020154  000425                                BR      NDONE
3267
3268 020156  000000                        NDAT00: .WORD   0
3269 020160  000000                        NDAT01: .WORD   0
3270 020162  000000                        NDAT02: .WORD   0
3271 020164  000000  052525  052525       NDAT03: .WORD   0,52525,52525,52525,52525
3272 020176  020220                        NPAT20: .WORD   NDAT10
3273 020200  070707                        NPAT21: .WORD   070707
3274 020202  070707                        NPAT22: .WORD   070707
3275 020204  070707  000001                NPAT23: .WORD   070707,1
3276 020210  177777                        NPAT10: .WORD   -1
3277 020212  177777                        NPAT11: .WORD   -1
3278 020214  177777                        NPAT12: .WORD   -1
3279 020216  177777                        NPAT13: .WORD   -1
3280
3281 020220  010421                        NDAT10: .WORD   010421
3282 020222  021042                        NDAT11: .WORD   021042
3283 020224  031463                        NDAT12: .WORD   031463
3284 020226  042104                        NDAT13: .WORD   042104
3285
3286 020230                                NDONE:
     020230  104413                                RSETUP                          ;GO INITIALIZE THE FPS AND STACK; AND
                                                                                    ;SEE IF THE USER HAS EXPRESSED
                                                                                    ;THE DESIRE TO CHANGE THE SOFTWARE
                                                                                    ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
                                                                                    ;THE USER TYPED CONTROL G?).
3287
3294
```

```
                                .SBTTL  TEST # 21 - FSRC MODE 5 TEST
                                ;;*****************************************************************
                                ;*TEST 21 - FSRC MODE 5 TEST
                                ;*
                                ;* THIS IS A TEST OF FSRC MODE 5, AUTO DECREMENT
                                ;* DEFERRED.
                                ;*
                                ;;*****************************************************************
020232  000004                TST21:  SCOPE
                                01:
020234                                  SETD                            ;SET FD MODE
020234  170011
                                        MOV     #OPAT10,R0
020236  012700  020714                  LDD     (R0),AC0                ;LOAD AC0 WITH A
020242  172410                          ;DEFAULT PATTERN.
                                        MOV     #OPAT21,R0
020244  012700  020702                  CLR     R3
020250  005003                          MOV     #OERR0,ERRVEC           ;IF A FAILURE
020252  012737  020424  000004          ;OCCURS IN THE FSRC
                                        ;FLOWS AN ODD ADDR.
                                        ;TRAP TO 4 MAY OCCUR.
020260  172450                  02:     LDD     @-(R0),AC0              ;TEST INSTRUCTION
020262  005203                  03:     INC     R3
020264  005203                  04:     INC     R3
                                        MOV     #ODAT00,R1
020266  012701  020662                  STD     AC0,(R1)                ;GET THE DATA
020272  174011
                                        CMP     R0,#OPAT20              ;WAS R0 DECREMENTED
020274  020027  020700                  BEQ     012                     ;BY 2?
020300  001436
                                05:     CMP     R0,#OPAT21+10           ;FSRC MODE 2
020302  020027  020712                  BNE     06
020306  001001                          BR      OERR1
020310  000505
                                06:     CMP     R0,#OPAT21-10           ;FSRC MODE 4?
020312  020027  020672                  BNE     07
020316  001001                          BR      OERR2
020320  000517
                                07:     CMP     R0,#OPAT21
020322  020027  020702
                                        MOV     #ODAT01,R2              ;FSRC MODE 0?
020326  012702  020664                  MOV     #4,R3
020332  012703  000004          08:     CMP     (R2)+,#-1
020336  022227  177777                  BNE     09
020342  001002                          SOB     R3,08
020344  077304                          BR      OERR3
020346  000510
                                09:     MOV     #ODAT00,R2              ;FSRC MODE 1?
020350  012702  020662                  MOV     #OPAT21,R3
020354  012703  020702                  MOV     #4,R4
020360  012704  000004          010:    CMP     (R2)+,(R3)+
020364  022223                          BNE     011
020366  001002                          SOB     R4,010
020370  077403                          BR      OERR4
020372  000502
```

```
3344 020374  000505                 011:    BR      OERR5
3345
3346 020376  012702  020662         012:    MOV     #ODAT00,R2      ;DATA CORRECT?
3347 020402  012703  020724                 MOV     #ODAT10,R3
3348 020406  012704  000004                 MOV     #4,R4
3349 020412  022223                 013:    CMP     (R2)+,(R3)+
3350 020414  001002                         BNE     014
3351 020416  077403                         SOB     R4,013
3352 020420  000545                         BR      ODONE
3353
3354 020422  000504                 014:    BR      OERR6
3355
3356                                 ;IF AN ODD ADDRESS TRAP OCCURS COME
3357                                 ;HERE TO SEE IF THE FAILURE WAS IN THE
3358                                 ;FSRC FLOWS:
3359
3360 020424  022716  020264         OERR0:  CMP     #04,(SP)        ;FSRC MODE 6 OR 7?
3361 020430  001412                         BEQ     OERR10
3362 020432  022716  020262                 CMP     #03,(SP)
3363 020436  001402                         BEQ     1$
3364 020440  000137  041174                 JMP     CPSPUR
3365 020444  020027  020704         1$:     CMP     R0,#OPAT21+2    ;FSRC MODE 3?
3366 020450  001425                         BEQ     OERR1
3367 020452  000137  041174                 JMP     CPSPUR
3368
3369 020456                         OERR10:                         ;WENT TO FSRC
3370 020456  011637  001236                 MOV     (SP),$TMP2      ;MODE 6 OR 7
3371 020462  022626                         CMP     (SP)+,(SP)+
3372 020464  104074                 1$:     ERROR   +74
3373 020466  000522                         BR      ODONE
3374
3375 020470  011637  001240         OERR11: MOV     (SP),$TMP3      ;WENT TO FSRC MODE
3376 020474  022626                         CMP     (SP)+,(SP)+     ;3
3377 020476  012737  000627  001244         MOV     #627,$TMP5
3378 020504  012737  000325  001250         MOV     #325,$TMP7
3379 020512  012737  000323  001246         MOV     #323,$TMP6
3380 020520  104075                 1$:     ERROR   +75
3381 020522  000504                         BR      ODONE
3382
3383 020524  012737  000322  001246 OERR1:  MOV     #322,$TMP6      ;FSRC MODE2
3384 020532  012737  000627  001242 OERR20: MOV     #627,$TMP4
3385 020540  012737  000325  001250         MOV     #325,$TMP7
3386 020546  012737  020260  001236         MOV     #02,$TMP2
3387 020554  104076                 1$:     ERROR   +76
3388 020556  000466                         BR      ODONE
3389 020560  012737  000324  001246 OERR2:  MOV     #324,$TMP6      ;FSRC MODE 4
3390 020566  000761                         BR      OERR20
3391 020570  012737  000320  001246 OERR3:  MOV     #320,$TMP6      ;FSRC MODE 0
3392 020576  000755                         BR      OERR20
3393 020600  012737  000321  001246 OERR4:  MOV     #321,$TMP6      ;FSRC MODE 1
3394 020606  000751                         BR      OERR20
3395
3396 020610  010037  001240         OERR5:  MOV     R0,$TMP3        ;R0 NOT DECREMENTED
3397 020614  012737  020700  001242         MOV     #OPAT20,$TMP4   ;PROPERLY
3398 020622  012737  020264  001236         MOV     #04,$TMP2
3399 020630  104077                 1$:     ERROR   +77
3400 020632  000440                         BR      ODONE
```

```
3401
3402 020634                                  OERR6:                                  ;DATA FAILURE
3403 020634  012737  020260  001236                  MOV     #02,$TMP2
3404 020642  012737  020662  001240                  MOV     #ODAT00,$TMP3
3405 020650  012737  020724  001242                  MOV     #ODATI0,$TMP4
3406 020656  104100                          1$:     ERROR   +100
3407 020660  000425                                  BR      ODONE
3408
3409 020662  000000                          ODAT00: .WORD   0
3410 020664  000000                          ODAT01: .WORD   0
3411 020666  000000                          ODAT02: .WORD   0
3412 020670  000000  052525  052525          ODAT03: .WORD   0,52525,52525,52525
3413 020700  020724                          OPAT20: .WORD   ODATI0
3414 020702  070707                          OPAT21: .WORD   070707
3415 020704  070707                          OPAT22: .WORD   070707
3416 020706  070707                          OPAT23: .WORD   070707
3417 020710  070707  000001                  OPAT24: .WORD   070707,1
3418 020714  177777                          OPAT10: .WORD   -1
3419 020716  177777                          OPAT11: .WORD   -1
3420 020720  177777                          OPAT12: .WORD   -1
3421 020722  177777                          OPAT13: .WORD   -1
3422
3423 020724  073567                          ODATI0: .WORD   73567
3424 020726  004210                          ODATI1: .WORD   004210
3425 020730  114631                          ODATI2: .WORD   114631
3426 020732  125252                          ODATI3: .WORD   125252
3427
3428 020734                                  ODONE:
     020734  104413                                  RSETUP                          ;GO INITIALIZE THE FPS AND STACK; AND
                                                                                     ;SEE IF THE USER HAS EXPRESSED
                                                                                     ;THE DESIRE TO CHANGE THE SOFTWARE
                                                                                     ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
                                                                                     ;THE USER TYPED CONTROL G?).
3429
3435
```

```
                                        .SBTTL  TEST # 22 - FSRC MODE 6 TEST
                                        ;;**********************************************************************
                                        ;*TEST 22 - FSRC MODE 6 TEST
                                        ;*
                                        ;* THIS IS A TEST OF FSRC MODE 6, INDEX MODE
                                        ;*
                                        ;;**********************************************************************
020736  000004                        TST22:  SCOPE
                                        P1:
020740  170011                                SETD                    ;SET FD MODE

020742  012700  021360                        MOV     #PPAT10,R0
020746  172410                                LDD     (R0),AC0        ;LOAD A DEFAULT PATTERN
                                                                      ;INTO AC0
020750  012737  021056  000004                MOV     #PERR0,ERRVECT  ;IF THE (BUT FSRC) FORQ
                                                                      ;FAILS AN ODD ADDRESS TRAP
020756  012700  021127                        MOV     #PDATI0-241,R0  ;COULD OCCUR.

020762  172460  000241                P2:     LDD     241(R0),AC0
        020764                          P3=P2+2

020766  012701  021400                P4:     MOV     #PDATO0,R1
020772  174011                                STD     AC0,(R1)        ;GET THE DATA
020774  012703  000004                        MOV     #4,R3
021000  012702  021370                        MOV     #PDATI0,R2
021004  012701  021400                        MOV     #PDATO0,R1
021010  022221                        P5:     CMP     (R2)+,(R1)+     ;CHECK THE DATA
021012  001007                                BNE     P6
021014  077303                                SOB     R3,P5
021016  022700  021127                        CMP     #PDATI0-241,R0  ;R0 CORRECT?
021022  001401                                BEQ     1$
021024  000512                                BR      PERR21
021026  000137  021410                1$:     JMP     PDONE

021032  012701  021400                P6:     MOV     #PDATO0,R1
021036  012703  000004                        MOV     #4,R3
021042  022721  177777                P7:     CMP     #-1,(R1)+       ;WAS IT FSRC MODE 0?
021046  001401                                BEQ     P8
021050  000512                                BR      PERR1
021052  077305                        P8:     SOB     R3,P7
021054  000523                                BR      PERR2
                                        ;TRAP TO HERE ON AN ODD ADDRESS
021056  021627  020764                PERR0:  CMP     (SP),#P3
021062  001411                                BEQ     PERR11
021064  021627  020766                        CMP     (SP),#P4        ;WAS IT FSRC MODE 7?
021070  001402                                BEQ     PERR10
021072  000137  041174                        JMP     CPSPUR

021076  012737  000327  001246        PERR10: MOV     #327,$TMP6
021104  000443                                BR      PERR17
021106  022700  021127                PERR11: CMP     #PDATI0-241,R0  ;WAS IT FSRC MODE 1
021112  001004                                BNE     PERR12
021114  012737  000321  001246                MOV     #321,$TMP6
021122  000434                                BR      PERR17
021124  022700  021137                PERR12: CMP     #PDATI0-241+10,R0       ;WAS IT FSRC MODE 2
021130  001004                                BNE     PERR13
```

```
3486 021132  012737  000322  001246          MOV     #322,$TMP6
3487 021140  000425                          BR      PERR17
3488 021142  022700  021131          PERR13: CMP     #PDATI0-241+2,R0        ;WAS IT FSRC MODE 3
3489 021146  001004                          BNE     PERR14
3490 021150  012737  000323  001246          MOV     #323,$TMP6
3491 021156  000416                          BR      PERR17
3492 021160  022700  021117          PERR14: CMP     #PDATI0-241-10,R0       ;WAS IT FSRC MODE 4
3493 021164  001004                          BNE     PERR15
3494 021166  012737  000324  001246          MOV     #324,$TMP6
3495 021174  000407                          BR      PERR17
3496 021176  022700  021125          PERR15: CMP     #PDATI0-241-2,R0        ;WAS IT FSRC MODE 5
3497 021202  001401                          BEQ     PERR16
3498 021204  000416                          BR      PERR20
3499 021206  012737  000325  001246  PERR16: MOV     #325,$TMP6
3500
3501 021214  012737  000627  001244  PERR17: MOV     #627,$TMP5      ;REPORT FSRC
3502 021222  012737  000326  001250          MOV     #326,$TMP7      ;FLOWS FAILURE.
3503 021230  011637  001236                  MOV     (SP),$TMP2
3504 021234  022626                          CMP     (SP)+,(SP)+
3505 021236  104101                  1$:     ERROR   +101
3506 021240  000463                          BR      PDONE
3507
3508 021242  011637  001236          PERR20: MOV     (SP),$TMP2      ;REPORT R0 AFFECTED
3509 021246  022626                          CMP     (SP)+,(SP)+
3510 021250  000403                          BR      PERR22
3511 021252  012737  020762  001236  PERR21: MOV     #P2,$TMP2
3512 021260                          PERR22:
3513 021260  010037  001240                  MOV     R0,$TMP3
3514 021264  012737  021127  001242          MOV     #PDATI0-241,$TMP4
3515 021272  104102                  1$:     ERROR   +102
3516 021274  000445                          BR      PDONE
3517
3518 021276                          PERR1:                          ;DATA FAILURE.
3519 021276  012737  020762  001236          MOV     #P2,$TMP2
3520 021304  012737  021370  001240          MOV     #PDATI0,$TMP3
3521 021312  012737  021400  001242          MOV     #PDAT00,$TMP4
3522 021320  104104                  1$:     ERROR   +104
3523 021322  000432                          BR      PDONE
3524
3525 021324                          PERR2:                          ;FSRC FAILURE TO
3526 021324  012737  020762  001236          MOV     #P2,$TMP2       ;MODE 0
3527 021332  012737  000627  001244          MOV     #627,$TMP5
3528 021340  012737  000326  001250          MOV     #326,$TMP7
3529 021346  012737  000320  001246          MOV     #320,$TMP6
3530 021354  104103                  1$:     ERROR   +103
3531 021356  000414                          BR      PDONE
3532
3533 021360  177777                  PPAT10: .WORD   -1
3534 021362  177777                  PPAT11: .WORD   -1
3535 021364  177777                  PPAT12: .WORD   -1
3536 021366  177777                  PPAT13: .WORD   -1
3537
3538 021370  010421                  PDATI0: .WORD   010421
3539 021372  031463                  PDATI1: .WORD   031463
3540 021374  052525                  PDATI2: .WORD   052525
3541 021376  073567                  PDATI3: .WORD   073567
3542
```

```
3543 021400  000000                PDAT00: .WORD   0
3544 021402  000000                PDAT01: .WORD   0
3545 021404  000000                PDAT02: .WORD   0
3546 021406  000000                PDAT03: .WORD   0
3547
3548 021410                        PDONE:
     021410  104413                        RSETUP                  ;GO INITIALIZE THE FPS AND STACK; AND
                                                                    ;SEE IF THE USER HAS EXPRESSED
                                                                    ;THE DESIRE TO CHANGE THE SOFTWARE
                                                                    ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
                                                                    ;THE USER TYPED CONTROL G?).
3549
3556
```

```
3557                                    .SBTTL  TEST # 23 - FSRC MODE 7 TEST
                                        ;;*****************************************************************
                                        ;*TEST 23 - FSRC MODE 7 TEST
                                        ;*
                                        ;* THIS IS A TEST OF FSRC MODE 7, INDEX
                                        ;* DEFERRED MODE.
                                        ;*
                                        ;;*****************************************************************
     021412  000004                     TST23:  SCOPE
3558
3559 021414                             Q1:
3560 021414  170011                             SETD
3561
3562 021416  012700  022050                     MOV     #QPAT10,R0
3563 021422  172410                             LDD     (R0),AC0        ;LOAD A DEFAULT
3564                                                                     ;PATTERN INTO AC0
3565 021424  012737  021556  000004             MOV     #QERR0,ERRVECT  ;IF THE (BUT FSRC)
3566                                                                     ;FORK FAILS AN
3567                                                                     ;ODD ADR TRAP COULD
3568                                                                     ;OCCUR
3569 021432  012700  021617                     MOV     #QPAT20-241,R0
3570
3571 021436  172470  000241             Q2:     LDD     @241(R0),AC0
3572          021440                     Q3=Q2+2
3573
3574 021442  012701  022070             Q4:     MOV     #QDAT00,R1
3575 021446  174011                             STD     AC0,(R1)        ;GET THE DATA
3576
3577 021450  012703  000004                     MOV     #4,R3
3578 021454  012704  022070                     MOV     #QDAT00,R4
3579 021460  012705  022100                     MOV     #QDAT10,R5
3580 021464  022425                     Q5:     CMP     (R4)+,(R5)+     ;CHECK THE DATA
3581 021466  001007                             BNE     Q6
3582 021470  077303                             SOB     R3,Q5
3583
3584 021472  022700  021617                     CMP     #QPAT20-241,R0  ;CHECK R0.
3585 021476  001401                             BEQ     1$
3586 021500  000514                             BR      QERR21
3587 021502  000137  022110             1$:     JMP     QDONE
3588
3589 021506  012701  022070             Q6:     MOV     #QDAT00,R1
3590 021512  012703  000004                     MOV     #4,R3
3591 021516  022721  177777             Q7:     CMP     #-1,(R1)+       ;WAS IT FSRC MODE 0?
3592 021522  001002                             BNE     Q8
3593 021524  077304                             SOB     R3,Q7
3594 021526  000513                             BR      QERR2
3595
3596 021530  012701  022060             Q8:     MOV     #QPAT20,R1
3597 021534  012702  022070                     MOV     #QDAT00,R2
3598 021540  012703  000004                     MOV     #4,R3
3599 021544  022122                     Q9:     CMP     (R1)+,(R2)+     ;WAS IT FSRC 6
3600 021546  001401                             BEQ     Q10             ;OR DATA FAILURE
3601 021550  000524                             BR      QERR1
3602 021552  077304                     Q10:    SOB     R3,Q9
3603 021554  000504                             BR      QERR3
3604
3605                                    ;TRAP TO HERE ON AN ODD ADR FAILURE
```

```
                                                
3606                                            
3607 021556  021627  020764          QERR0:  CMP     (SP),#P3
3608 021562  000137  041174                  JMP     CPSPUR
3609                                            
3610 021566  022700  021617          QERR11: CMP     #QPAT20-241,R0  ;WAS IT FSRC
3611 021572  001004                          BNE     QERR12          ;MODE 1?
3612 021574  012737  000321  001246          MOV     #321,$TMP6
3613 021602  000434                          BR      QERR17
3614 021604  022700  021627          QERR12: CMP     #QPAT20-241+10,R0       ;WAS IT FSRC
3615 021610  001004                          BNE     QERR13          ;MODE 2?
3616 021612  012737  000322  001246          MOV     #322,$TMP6
3617 021620  000425                          BR      QERR17
3618 021622  022700  021621          QERR13: CMP     #QPAT20-241+2,R0        ;WAS IT FSRC
3619 021626  001004                          BNE     QERR14          ;MODE 3?
3620 021630  012737  000323  001246          MOV     #323,$TMP6
3621 021636  000416                          BR      QERR17
3622 021640  022700  021607          QERR14: CMP     #QPAT20-241-10,R0       ;WAS IT FSRC
3623 021644  001004                          BNE     QERR15          ;MODE 4
3624 021646  012737  000324  001246          MOV     #324,$TMP6
3625 021654  000407                          BR      QERR17
3626                                            
3627 021656  022700  021615          QERR15: CMP     #QPAT20-241-2,R0        ;WAS IT FSRC
3628 021662  001401                          BEQ     QERR16          ;MODE 5
3629 021664  000416                          BR      QERR20
3630                                            
3631 021666  012737  000325  001246  QERR16: MOV     #325,$TMP6
3632                                            
3633 021674  012737  000627  001244  QERR17: MOV     #627,$TMP5      ;REPORT FSRC FAILURE
3634 021702  012737  000327  001250          MOV     #327,$TMP7
3635 021710  011637  001236                  MOV     (SP),$TMP2
3636 021714  022626                          CMP     (SP)+,(SP)+
3637 021716  104105                  1$:     ERROR   +105
3638 021720  000473                          BR      QDONE
3639                                            
3640 021722  011637  001236          QERR20: MOV     (SP),$TMP2      ;REPORT R0 AFFECTED.
3641 021726  022626                          CMP     (SP)+,(SP)+
3642 021730  000403                          BR      QERR22
3643 021732  012737  021436  001236  QERR21: MOV     #Q2,$TMP2
3644 021740                          QERR22:
3645 021740  010037  001240                  MOV     R0,$TMP3
3646 021744  012737  021617  001242          MOV     #QPAT20-241,$TMP4
3647 021752  104106                  1$:     ERROR   +106
3648 021754  000455                          BR      QDONE
3649                                            
3650 021756  012737  000320  001246  QERR2:  MOV     #320,$TMP6      ;WENT TO FSRC
3651 021764  000403                          BR      QERR4           ;MODE 0
3652 021766  012737  000326  001246  QERR3:  MOV     #326,$TMP6      ;WENT TO FSRC
3653                                                                  ;MODE 6
3654 021774  012737  000627  001244  QERR4:  MOV     #627,$TMP5
3655 022002  012737  000327  001250          MOV     #327,$TMP7
3656 022010  012737  021436  001236          MOV     #Q2,$TMP2
3657 022016  104107                  1$:     ERROR   +107
3658 022020  000433                          BR      QDONE
3659                                            
3660 022022                          QERR1:                          ;DATA FAILURE
3661 022022  012737  021436  001236          MOV     #Q2,$TMP2
3662 022030  012737  022100  001240          MOV     #QDATI0,$TMP3
```

```
3663 022036  012737  022070  001242          MOV     #QDAT00,$TMP4
3664 022044  104110                  1$:     ERROR   +110
3665 022046  000420                          BR      QDONE
3666
3667 022050  177777                  QPAT10: .WORD   -1
3668 022052  177777                  QPAT11: .WORD   -1
3669 022054  177777                  QPAT12: .WORD   -1
3670 022056  177777                  QPAT13: .WORD   -1
3671
3672 022060  022100                  QPAT20: .WORD   QDAT10
3673 022062  052525                  QPAT21: .WORD   52525
3674 022064  052525                  QPAT22: .WORD   52525
3675 022066  052525                  QPAT23: .WORD   52525
3676
3677 022070  000000                  QDAT00: .WORD   0
3678 022072  000000                  QDAT01: .WORD   0
3679 022074  000000                  QDAT02: .WORD   0
3680 022076  000000                  QDAT03: .WORD   0
3681
3682 022100  073567                  QDAT10: .WORD   073567
3683 022102  052525                  QDAT11: .WORD   052525
3684 022104  031463                  QDAT12: .WORD   031463
3685 022106  010421                  QDAT13: .WORD   010421
3686
3687 022110                           QDONE:
     022110  104413                          RSETUP                  ;GO INITIALIZE THE FPS AND STACK; AND
                                                                     ;SEE IF THE USER HAS EXPRESSED
                                                                     ;THE DESIRE TO CHANGE THE SOFTWARE
                                                                     ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
                                                                     ;THE USER TYPED CONTROL G?).
3688
```

TEST # 24 - (BUT EZBT Y8),(BUT ENBT) AND (BUT FIUV) TEST

```
3705                                    .SBTTL  TEST # 24 - (BUT EZBT Y8),(BUT ENBT) AND (BUT FIUV) TEST
                                        ;;*****************************************************************
                                        ;*TEST 24 - (BUT EZBT Y8),(BUT ENBT) AND (BUT FIUV) TEST
                                        ;*
                                        ;* THIS IS A TEST OF THE (BUT EZBT Y8) FORK, THE
                                        ;* (BUT ENBT) FORK AND (BUT FIUV) FORK IN THE
                                        ;* LOAD INSTRUCTION FLOWS.
                                        ;* EACH OF THE PATTERNS:
                                        ;*              0
                                        ;*              +NUM
                                        ;*              -NUM
                                        ;*              -0
                                        ;* IS LOADED TWICE, ONCE WITH AC>0 THEN
                                        ;* WITH AC=0.  AFTER EACH LOAD THE FPS IS
                                        ;* CHECK TO INSURE THAT CONTROL WAS PASSED
                                        ;* THROUGH WITH THE FORKS PROPERLY.
                                        ;*
                                        ;;*****************************************************************
    022112  000004                      TST24:  SCOPE
3706 022114  005037  023236                     CLR     UFLAG
3707 022120  012700  023166                     MOV     #UPAT00,R0      ;SET UP AC#0 DATA.
3708 022124  012701  000004                     MOV     #4,R1
3709 022130  012720  177777             U0:     MOV     #-1,(R0)+
3710 022134  077103                             SOB     R1,U0
3711
3712 022136  012737  000033  023240             MOV     #033,UTMP1
3713 022144  012737  000023  023242             MOV     #023,UTMP2
3714 022152  012737  022716  000244             MOV     #UERR0,FPVECT   ;IN CASE (BUT FIUV FAILS)
3715 022160                             U1:
    022160  012737  022166  001110             MOV     #1$,$LPERR      ;SET UP THE LOOP ON ERROR ADDRESS.
3716 022166  012700  000200             1$:     MOV     #200,R0
3717 022172  170100                             LDFPS   R0
3718 022174  012700  023166                     MOV     #UPAT00,R0      ;LOAD AC0
3719 022200  172410                             LDD     (R0),AC0
3720 022202  013737  023240  023244             MOV     UTMP1,UROM1
3721 022210  012737  000001  023246             MOV     #001,UROM2
3722 022216  012737  000254  023250             MOV     #254,UROM3
3723
3724 022224  012700  023176                     MOV     #UPAT10,R0      ;LOAD 0 INTO AC0
3725 022230  172410                     U2:     LDD     (R0),AC0
3726 022232  010037  001252                     MOV     R0,$TMP10
3727 022236  012737  022230  001236             MOV     #U2,$TMP2
3728
3729 022244  012704  000204                     MOV     #204,R4         ;SEE IF FPS IS CORRECT
3730 022250  170205                             STFPS   R5
3731
3732 022252  020405                             CMP     R4,R5
3733 022254  001402                             BEQ     U3
3734 022256  000137  022742                     JMP     UERR1
3735
3736 022262                             U3:
    022262  012737  022270  001110             MOV     #1$,$LPERR      ;SET UP THE LOOP ON ERROR ADDRESS.
3737 022270  012700  000200             1$:     MOV     #200,R0
3738 022274  170100                             LDFPS   R0
3739
3740 022276  012700  023166                     MOV     #UPAT00,R0      ;LOAD AC0
3741 022302  172410                             LDD     (R0),AC0
```

```
3742 022304  013737  023242  023244          MOV     UTMP2,UROM1
3743 022312  012737  000003  023246          MOV     #003,UROM2
3744 022320  012737  000054  023250          MOV     #054,UROM3
3745
3746 022326  012700  023206                  MOV     #UPAT20,R0      ;LOAD A POSITIVE NUMBER
3747                                                                 ;INTO AC0
3748 022332  172410                  U4:     LDD     (R0),AC0
3749 022334  010037  001252                  MOV     R0,$TMP10
3750 022340  012737  022332  001236          MOV     #U4,$TMP2
3751 022346  012704  000200                  MOV     #200,R4         ;FPS CORRECT?
3752 022352  170205                        STFPS   R5
3753 022354  020405                        CMP     R4,R5
3754 022356  001402                        BEQ     U5
3755 022360  000137  023026                JMP     UERR2
3756 022364                          U5:
     022364  012737  022372  001110          MOV     #1$,$LPERR      ;SET UP THE LOOP ON ERROR ADDRESS.
3757 022372  012700  000200          1$:     MOV     #200,R0
3758 022376  170100                        LDFPS   R0
3759 022400  012700  023166                  MOV     #UPAT00,R0      ;LOAD AC0
3760 022404  172410                        LDD     (R0),AC0
3761 022406  013737  023242  023244          MOV     UTMP2,UROM1
3762 022414  012737  000403  023246          MOV     #403,UROM2
3763 022422  012737  000056  023250          MOV     #056,UROM3
3764 022430  012700  023216                  MOV     #UPAT30,R0      ;LOAD A NEGATIUE
3765                                                                 ;NUMBER INTO AL0
3766 022434  172410                  U6:     LDD     (R0),AC0
3767 022436  010037  001252                  MOV     R0,$TMP10
3768 022442  012737  022434  001236          MOV     #U6,$TMP2
3769 022450  012704  000210                  MOV     #210,R4         ;FPS CORRECT
3770 022454  170205                        STFPS   R5
3771 022456  020405                        CMP     R4,R5
3772 022460  001402                        BEQ     U7
3773 022462  000137  023026                JMP     UERR2
3774 022466                          U7:
     022466  012737  022474  001110          MOV     #1$,$LPERR      ;SET UP THE LOOP ON ERROR ADDRESS.
3775 022474  012700  000200          1$:     MOV     #200,R0
3776 022500  170100                        LDFPS   R0
3777 022502  012700  023166                  MOV     #UPAT00,R0      ;LOAD AC0
3778 022506  172410                        LDD     (R0),AC0
3779 022510  013737  023240  023244          MOV     UTMP1,UROM1
3780 022516  012737  000401  023246          MOV     #401,UROM2
3781 022524  012737  000256  023250          MOV     #256,UROM3
3782 022532  012700  023226                  MOV     #UPAT40,R0      ;LOAD -0 INTO AC0
3783 022536  172410                  U10:    LDD     (R0),AC0
3784 022540  000240                  U11:    NOP                     ;TRAP FROM HERE IF
3785 022542  010037  001252                  MOV     R0,$TMP10
3786 022546  012737  022536  001236          MOV     #U10,$TMP2      ;(BUT FIUV) FAULS!
3787 022554  012704  000214                  MOV     #214,R4         ;SEE IF FPS IS CORRECT.
3788 022560  170205                        STFPS   R5
3789 022562  020405                        CMP     R4,R5
3790 022564  001402                        BEQ     U12
3791 022566  000137  022742                JMP     UERR1
3792 022572  005737  023236          U12:    TST     UFLAG           ;SEE IF ALL THE PATTERNS
3793 022576  001021                        BNE     U14             ;HAVE BEEN TEST WITH
3794                                                                 ;BOTH AC NOT EQUAL TO 0 AND AC=0
3795 022600  012700  023166                  MOV     #UPAT00,R0      ;IF NOT GO BACK AND
3796 022604  012701  000004                  MOV     #4,R1           ;CHECK THEM WITH AC=0
```

```
                                U13:    CLR     (R0)+
                                        SOB     R1,U13
                                        MOV     #-1,UFLAG
                                        MOV     #233,UTMP1
                                        MOV     #223,UTMP2
                                        JMP     U1
                                U14:
                                        MOV     #1$,$LPERR      ;SET UP THE LOOP ON ERROR ADDRESS.
                                ;NOW SEE IF A TRAP CAN BE FORCED BY SETTING FIUV AND LOADING -0
                                1$:     MOV     #UERR3,FPVECT
                                        MOV     #4200,R0        ;SET FD AND FIUV
                                        LDFPS   R0
                                        MOV     #UPAT00,R0      ;SET UP AC0
                                        LDD     (R0),AC0
                                        MOV     #UPAT40,R0      ;LOAD -0
                                U15:    LDD     (R0),AC0        ;SHOULD TRAP TO 244
                                U16:    CFCC
                                        NOP
                                        MOV     #U15,$TMP2      ;REPORT ERROR.
                                                                ;DIDN'T TRAP
                                1$:     ERROR   +127            ;(BUT FIUV) FAILED.
                                        BR      UDONE

                                ;TRAPPED TO 244.  DID (BUT FIUV) FAIL?
                                UERR0:  CMP     (SP),#U11
                                        BEQ     1$
                                        JMP     FPSPUR
                                1$:     MOV     (SP),$TMP2
                                        CMP     (SP)+,(SP)+
                                2$:     ERROR   +126
                                        BR      UDONE

                                ;COME HERE TO ANALYZE FPS ERRORS

                                UERR1:  BIT     #4,R5
                                        BEQ     UERR20
                        001244  UERR10: MOV     #443,$TMP5
                                        MOV     UROM3,R3
                                        MOV     R3,$TMP7
                                        BIT     #200,R3
                                        BEQ     1$
                                        BIC     #200,R3
                                        BR      2$
                                1$:     BIS     #200,R3
                                2$:     MOV     R3,$TMP6
                                UERR11: MOV     R5,$TMP3
                                        MOV     R4,$TMP4
                                1$:     ERROR   +124
                                        BR      UDONE
                                UERR2:  BIT     #4,R5
                                        BEQ     UERR10
                        001244  UERR20: MOV     UROM1,$TMP5
                                        MOV     UROM2,R3
                                        MOV     R3,$TMP7
                                        BIT     #400,R3
                                        BEQ     1$
                                        BIC     #400,R3
```

```
3853 023064  000402                          BR      2$
3854 023066  052703  000400          1$:     BIS     #400,R3
3855 023072  010337  001246          2$:     MOV     R3,$TMP6
3856 023076  010537  001240          UERR21: MOV     R5,$TMP3
3857 023102  010437  001242                  MOV     R4,$TMP4
3858 023106  104125                  1$:     ERROR   +125
3859 023110  000460                          BR      UDONE
3860
3861                                 ;INTERRUPT HERE WHEN FIUV SET AND ATTEMPTED TO LOAD-0
3862 023112  021627  022700          UERR3:  CMP     (SP),#U16
3863 023116  001402                          BEQ     1$
3864 023120  000137  041142                  JMP     FPSPUR
3865 023124  022626                  1$:     CMP     (SP)+,(SP)+
3866 023126  005000                          CLR     R0
3867 023130  170300                          STST    R0                      ;GET FEC.
3868 023132  022700  000014                  CMP     #14,R0                  ;CORRECT
3869 023136  001001                          BNE     UERR4
3870 023140  000444                          BR      UDONE
3871 023142  012737  022676  001236  UERR4:  MOV     #U15,$TMP2
3872 023150  012737  000012  001242          MOV     #12,$TMP4
3873 023156  010037  001240                  MOV     R0,$TMP3
3874 023162  104130                  1$:     ERROR   +130
3875 023164  000432                          BR      UDONE
3876 023166  000000                  UPAT00: .WORD   0
3877 023170  000000                  UPAT01: .WORD   0
3878 023172  000000                  UPAT02: .WORD   0
3879 023174  000000                  UPAT03: .WORD   0
3880
3881 023176  000000                  UPAT10: .WORD   0                       ;0
3882 023200  000000                  UPAT11: .WORD   0
3883 023202  000000                  UPAT12: .WORD   0
3884 023204  000000                  UPAT13: .WORD   0
3885
3886 023206  010421                  UPAT20: .WORD   010421                  ;POS NUM
3887 023210  114631                  UPAT21: .WORD   114631
3888 023212  125252                  UPAT22: .WORD   125252
3889 023214  177777                  UPAT23: .WORD   177777
3890
3891 023216  114631                  UPAT30: .WORD   114631                  ;NEG NUM
3892 023220  135673                  UPAT31: .WORD   135673
3893 023222  146314                  UPAT32: .WORD   146314
3894 023224  167356                  UPAT33: .WORD   167356
3895
3896 023226  100000                  UPAT40: .WORD   100000                  ;NEG ZERO
3897 023230  000000                  UPAT41: .WORD   0
3898 023232  000000                  UPAT42: .WORD   0
3899 023234  000000                  UPAT43: .WORD   0
3900
3901 023236  000000                  UFLAG:  .WORD   0
3902 023240  000000                  UTMP1:  .WORD   0
3903 023242  000000                  UTMP2:  .WORD   0
3904 023244  000000                  UROM1:  .WORD   0
3905 023246  000000                  UROM2:  .WORD   0
3906 023250  000000                  UROM3:  .WORD   0
3907 023252                          UDONE:
3908
3909
```

```
                                                                J 8
3917                                    .SBTTL  TEST # 25 - ADDF,ADDD,SUBF AND SUBD WITH FSRC=AC=0 TEST
                                        ;;*****************************************************************
                                        ;*TEST 25 - ADDF,ADDD,SUBF AND SUBD WITH FSRC=AC=0 TEST

                                        ;*
                                        ;* THIS IS A TEST OF ADD AND SUB WITH FSRC=AC=0
                                        ;*

                                        ;;*****************************************************************
 023252  000004                         TST25:  SCOPE
3918 023254
 023254  012737  023262  001110         W1:     MOV     #1$,$LPERR      ;SET UP THE LOOP ON ERROR ADDRESS.
3919 023262  012700  000200             1$:     MOV     #200,R0
3920 023266  170100                             LDFPS   R0              ;SET DOUBLE MODE
3921 023270  012700  024026                     MOV     #WPAT00,R0      ;LOAD AC0=:
3922 023274  172410                             LDD     (R0),AC0
3923 023276  012737  023310  001236             MOV     #W2,$TMP2
3924 023304  012700  024026                     MOV     #WPAT00,R0
3925 023310  172010                     W2:     ADDD    (R0),AC0        ;TEST INSTRUCTION.
3926 023312  170205                             STFPS   R5              ;GET FPS
3927 023314  170011                             SETD                    ;SET DOUBLE MODE
3928 023316  012700  024026                     MOV     #WPAT00,R0
3929 023322  174010                             STD     AC0,(R0)        ;GET THE RESULT
3930 023324  012701  024026                     MOV     #WPAT00,R1
3931 023330  012702  000004                     MOV     #4,R2
3932 023334  022021                     W3:     CMP     (R0)+,(R1)+     ;IS RESULT CORRECT
3933 023336  001405                             BEQ     W4
3934                                                                    ;NO
3935 023340  004737  023774                     JSR     PC,WSETUP
3936 023344  104133                     1$:     ERROR   +133
3937 023346  000137  024046                     JMP     WDONE
3938 023352  077210                     W4:     SOB     R2,W3
3939 023354  022705  000204                     CMP     #204,R5         ;IS FPS CORRECT
3940 023360  001410                             BEQ     W5
3941                                                                    ;NO
3942 023362  012737  000204  001242             MOV     #204,$TMP4
3943 023370  010537  001240                     MOV     R5,$TMP3
3944 023374  104137                     1$:     ERROR   +137
3945 023376  000137  024046                     JMP     WDONE
3946 023402                             W5:
 023402  012737  023410  001110         W5:     MOV     #1$,$LPERR      ;SET UP THE LOOP ON ERROR ADDRESS.
3947 023410  012700  000200             1$:     MOV     #200,R0
3948 023414  170100                             LDFPS   R0              ;SET DOUBLE MODE
3949 023416  012700  024026                     MOV     #WPAT00,R0      ;LOAD AC0=0
3950 023422  172410                             LDD     (R0),AC0
3951 023424  012737  023442  001236             MOV     #W6,$TMP2
3952 023432  005000                             CLR     R0
3953 023434  170100                             LDFPS   R0              ;GO TO FLOATING MODE
3954 023436  012700  024026                     MOV     #WPAT00,R0
3955 023442  172010                     W6:     ADDF    (R0),AC0        ;TEST INSTRUCTION
3956 023444  170205                             STFPS   R5              ;GET FPS
3957 023446  170011                             SETD                    ;RESET TO DOUBLE MODE
3958 023450  012700  024026                     MOV     #WPAT00,R0
3959 023454  174010                             STD     AC0,(R0)        ;GET THE RESULT
3960 023456  012701  024026                     MOV     #WPAT00,R1
3961 023462  012702  000004                     MOV     #4,R2
3962 023466  022021                     W7:     CMP     (R0)+,(R1)+     ;WAS THE RESULT
```

```
3963 023470  001402                    BEQ     W10             ;NO. REPORT FAILURE.
3964 023472  104134            1$:     ERROR   +134
3965 023474  000564                    BR      WDONE
3966 023476  077205            W10:    SOB     R2,W7
3967 023500  022705  000004            CMP     #4,R5           ;WAS FPS CORRECT
3968 023504  001407                    BEQ     W11
3969                                                            ;INCORRECT FPS.
3970 023506  012737  000004  001242    MOV     #4,$TMP4
3971 023514  010537  001240            MOV     R5,$TMP3
3972 023520  104140            1$:     ERROR   +140
3973 023522  000551                    BR      WDONE
3974 023524                    W11:
     023524  012737  023532  001110    MOV     #1$,$LPERR      ;SET UP THE LOOP ON ERROR ADDRESS.
3975 023532  012700  000200    1$:     MOV     #200,R0
3976 023536  170100                    LDFPS   R0              ;SET DOUBLE MODE
3977 023540  012700  024026            MOV     #WPAT00,R0      ;LOAD AC0=0
3978 023544  172410                    LDD     (R0),AC0
3979 023546  012737  023560  001236    MOV     #W12,$TMP2
3980 023554  012700  024026            MOV     #WPAT00,R0
3981 023560  173010            W12:    SUBD    (R0),AC0        ;TEST INSTRUCTION
3982 023562  170205                    STFPS   R5              ;GET FPS
3983 023564  170011                    SETD                    ;SET DOUBLE MODE
3984 023566  012700  024026            MOV     #WPAT00,R0
3985 023572  174010                    STD     AC0,(R0)        ;GET THE RESULT
3986 023574  012701  024026            MOV     #WPAT00,R1
3987 023600  012702  000004            MOV     #4,R2
3988 023604  022021            W13:    CMP     (R0)+,(R1)+     ;IS RESULT CORRECT?
3989 023606  001404                    BEQ     W14
3990                                                            ;NO.
3991 023610  004737  023774            JSR     PC,WSETUP
3992 023614  104135            1$:     ERROR   +135
3993 023616  000513                    BR      WDONE
3994 023620  077207            W14:    SOB     R2,W13
3995 023622  022705  000204            CMP     #204,R5         ;IS FPS CORRECT?
3996 023626  001407                    BEQ     W15
3997                                                            ;NO.
3998 023630  012737  000204  001242    MOV     #204,$TMP4
3999 023636  010537  001240            MOV     R5,$TMP3
4000 023642  104141            1$:     ERROR   +141
4001 023644  000500                    BR      WDONE
4002 023646                    W15:
     023646  012737  023654  001110    MOV     #1$,$LPERR      ;SET UP THE LOOP ON ERROR ADDRESS.
4003 023654  012700  000200    1$:     MOV     #200,R0
4004 023660  170100                    LDFPS   R0              ;SET DOUBLE MODE
4005 023662  012700  024026            MOV     #WPAT00,R0      ;LOAD AC0=0
4006 023666  172410                    LDD     (R0),AC0
4007 023670  012737  023706  001236    MOV     #W16,$TMP2
4008 023676  005000                    CLR     R0
4009 023700  170100                    LDFPS   R0              ;ENTER FLOATING MODE.
4010 023702  012700  024026            MOV     #WPAT00,R0
4011 023706  173010            W16:    SUBF    (R0),AC0        ;TEST INSTRUCTION.
4012 023710  170205                    STFPS   R5              ;GET FPS
4013 023712  170011                    SETD                    ;RESET TO DOUBLE MODE
4014 023714  012700  024026            MOV     #WPAT00,R0      ;GET THE RESULT.
4015 023720  174010                    STD     AC0,(R0)
4016 023722  012701  024026            MOV     #WPAT00,R1
4017 023726  012702  000004            MOV     #4,R2
```

```
4018 023732  022021                  W17:    CMP     (R0)+,(R1)+     ;IS RESULT CORRECT?
4019 023734  001404                          BEQ     W20
4020                                                                 ;NO.
4021 023736  004737  023774                  JSR     PC,WSETUP
4022 023742  104136                  1$:     ERROR   +136
4023 023744  000440                          BR      WDONE
4024 023746  077207                  W20:    SOB     R2,W17
4025 023750  022705  000004                  CMP     #4,R5           ;IS FPS CORRECT?
4026 023754  001434                          BEQ     WDONE
4027                                                                 ;NO
4028 023756  012737  000004  001242          MOV     #4,$TMP4
4029 023764  010537  001240                  MOV     R5,$TMP3
4030 023770  104142                  1$:     ERROR   +142
4031 023772  000425                          BR      WDONE
4032
4033                                 ;SET UP FOR ERROR CALL
4034
4035 023774  012737  024026  001240  WSETUP: MOV     #WPAT00,$TMP3
4036 024002  012737  024026  001242          MOV     #WPAT00,$TMP4
4037 024010  012737  024026  001246          MOV     #WPAT00,$TMP6
4038 024016  012737  024026  001244          MOV     #WPAT00,$TMP5
4039 024024  000207                          RTS     PC
4040 024026  000000                  WPAT00: .WORD   0
4041 024030  000000                  WPAT01: .WORD   0
4042 024032  000000                  WPAT02: .WORD   0
4043 024034  000000                  WPAT03: .WORD   0
4044
4045 024036  000000                  WDAP00: .WORD   0
4046 024040  000000                  WDAT01: .WORD   0
4047 024042  000000                  WDAT02: .WORD   0
4048 024044  000000                  WDAT03: .WORD   0
4049
4050 024046                          WDONE:
     024046  104413                          RSETUP                  ;GO INITIALIZE THE FPS AND STACK; AND
                                                                     ;SEE IF THE USER HAS EXPRESSED
                                                                     ;THE DESIRE TO CHANGE THE SOFTWARE
                                                                     ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
                                                                     ;THE USER TYPED CONTROL G?).
4051
4052
```

```
                                        .SBTTL  TEST # 26 - ADDD AND SUB WITH FSRC=0
                                        ;;*********************************************************************
                                        ;*TEST 26 - ADDD AND SUB WITH FSRC=0
                                        ;*
                                        ;* THIS IS A TEST OF ADD AND SUB WITH FSRC=0.
                                        ;*
                                        ;;*********************************************************************
 024050  000004                         TST26:  SCOPE
4060 024052                             X1:
 024052  012737  024060  001110                 MOV     #1$,$LPERR      ;SET UP THE LOOP ON ERROR ADDRESS.
4061 024060  012700  000200             1$:     MOV     #200,R0
4062 024064  170100                             LDFPS   R0              ;SET DOUBLE MODE
4063 024066  012700  024632                     MOV     #XPAT00,R0      ;SET AC0 TO POSITIVE
4064 024072  010037  024620                     MOV     R0,XTMP         ;NUMBER #0
4065 024076  172410                             LDD     (R0),AC0
4066 024100  012737  024112  001236             MOV     #X2,$TMP2
4067 024106  012700  024642                     MOV     #XPAT10,R0      ;FSRC=0
4068 024112  172010                     X2:     ADDD    (R0),AC0        ;TEST INSTRUCTION
4069 024114  170205                             STFPS   R5
4070 024116  170011                             SETD
4071 024120  012700  024622                     MOV     #XDAT00,R0      ;GET RESULT.
4072 024124  174010                             STD     AC0,(R0)
4073 024126  012701  024632                     MOV     #XPAT00,R1
4074 024132  012702  000004                     MOV     #4,R2
4075 024136  022021                     X3:     CMP     (R0)+,(R1)+     ;IS RESULT CORRECT?
4076 024140  001401                             BEQ     X4
4077 024142  000561                             BR      XERR1
4078 024144  077204                     X4:     SOB     R2,X3
4079 024146  012704  000200                     MOV     #200,R4
4080 024152  020405                             CMP     R4,R5           ;IS FPS CORRECT?
4081 024154  001402                             BEQ     X5
4082 024156  000137  024570                     JMP     XERR2
4083 024162                             X5:
 024162  012737  024170  001110                 MOV     #1$,$LPERR      ;SET UP THE LOOP ON ERROR ADDRESS.
4084 024170  012700  000200             1$:     MOV     #200,R0
4085 024174  170100                             LDFPS   R0              ;SET DOUBLE MODE
4086 024176  012700  024652                     MOV     #XPAT20,R0      ;SET AC0 TO
4087 024202  010037  024620                     MOV     R0,XTMP         ;NEGATIVE NUMBER
4088 024206  172410                             LDD     (R0),AC0
4089 024210  012737  024222  001236             MOV     #X6,$TMP2
4090 024216  012700  024642                     MOV     #XPAT10,R0      ;FSRC=0
4091 024222  172010                     X6:     ADDD    (R0),AC0        ;TEST INSTRUCTION
4092 024224  170205                             STFPS   R5
4093 024226  170011                             SETD
4094 024230  012700  024622                     MOV     #XDAT00,R0      ;GET RESULT
4095 024234  174010                             STD     AC0,(R0)
4096 024236  012701  024652                     MOV     #XPAT20,R1
4097 024242  012702  000004                     MOV     #4,R2
4098 024246  022021                     X7:     CMP     (R0)+,(R1)+     ;IS RESULT CORRECT?
4099 024250  001401                             BEQ     X10
4100 024252  000515                             BR      XERR1
4101 024254  077204                     X10:    SOB     R2,X7
4102 024256  012704  000210                     MOV     #210,R4
4103 024262  020405                             CMP     R4,R5           ;IS FPS CORRECT?
4104 024264  001401                             BEQ     X11
4105 024266  000540                             BR      XERR2
```

```
4106 024270                                X11:
     024270  012737  024276  001110              MOV     #1$,$LPERR      ;SET UP THE LOOP ON ERROR ADDRESS.
4107 024276  012700  000200          1$:         MOV     #200,R0
4108 024302  170100                              LDFPS   R0              ;SET DOUBLE MODE
4109 024304  012700  024632                      MOV     #XPAT00,R0      ;SET AC0 TO NON-ZERO
4110 024310  010037  024620                      MOV     R0,XTMP         ;POSITIVE NUMBER
4111 024314  172410                              LDD     (R0),AC0
4112 024316  012737  024330  001236              MOV     #X12,$TMP2
4113 024324  012700  024642                      MOV     #XPAT10,R0      ;FSRC=0
4114 024330  173010                  X12:        SUBD    (R0),AC0        ;TEST INSTRUCTION
4115 024332  170205                              STFPS   R5
4116 024334  170011                              SETD
4117 024336  012700  024622                      MOV     #XDAT00,R0      ;GET RESULT
4118 024342  174010                              STD     AC0,(R0)
4119 024344  012701  024632                      MOV     #XPAT00,R1
4120 024350  012702  000004                      MOV     #4,R2
4121 024354  022021                  X13:        CMP     (R0)+,(R1)+     ;IS RESULT CORRECT?
4122 024356  001401                              BEQ     X14
4123 024360  000465                              BR      XERR3
4124 024362  077204                  X14:        SOB     R2,X13
4125 024364  012704  000200                      MOV     #200,R4         ;IS FPS CORRECT?
4126 024370  020405                              CMP     R4,R5
4127 024372  001401                              BEQ     X15
4128 024374  000503                              BR      XERR4
4129 024376                                X15:
     024376  012737  024404  001110              MOV     #1$,$LPERR      ;SET UP THE LOOP ON ERROR ADDRESS.
4130 024404  012700  000200          1$:         MOV     #200,R0
4131 024410  170100                              LDFPS   R0              ;SET DOUBLE MODE
4132 024412  012700  024652                      MOV     #XPAT20,R0      ;SET AC0=A NEGATIVE
4133 024416  010037  024620                      MOV     R0,XTMP         ;NUMBER
4134 024422  172410                              LDD     (R0),AC0
4135 024424  012737  024436  001236              MOV     #X16,$TMP2
4136 024432  012700  024642                      MOV     #XPAT10,R0      ;FSRC=0
4137 024436  173010                  X16:        SUBD    (R0),AC0        ;TEST INSTRUCTION.
4138 024440  170205                              STFPS   R5
4139 024442  170011                              SETD
4140 024444  012700  024622                      MOV     #XDAT00,R0      ;GET RESULT
4141 024450  174010                              STD     AC0,(R0)
4142 024452  012701  024652                      MOV     #XPAT20,R1
4143 024456  012702  000004                      MOV     #4,R2
4144 024462  022021                  X17:        CMP     (R0)+,(R1)+     ;IS RESULT CORRECT?
4145 024464  001401                              BEQ     X20
4146 024466  000422                              BR      XERR3
4147 024470  077204                  X20:        SOB     R2,X17
4148 024472  012704  000210                      MOV     #210,R4         ;IS FPS CORRECT?
4149 024476  020405                              CMP     R4,R5
4150 024500  001401                              BEQ     X21
4151 024502  000440                              BR      XERR4
4152 024504  000466                  X21:        BR      XDONE
4153
4154                                     ;REPORT DATA ERRORS
4155
4156 024506  012737  024642  001240  XERR1:     MOV     #XPAT10,$TMP3
4157 024514  013737  024620  001242              MOV     XTMP,$TMP4
4158 024522  012737  024622  001244              MOV     #XDAT00,$TMP5
4159 024530  104143                  1$:         ERROR   +143
4160 024532  000453                              BR      XDONE
```

```
4161 024534  012737  024642  001240  XERR3:  MOV     #XPAT10,$TMP3
4162 024542  013737  024620  001242          MOV     XTMP,$TMP4
4163 024550  012737  024622  001244          MOV     #XDAT00,$TMP5
4164 024556  013737  024620  001246          MOV     XTMP,$TMP6
4165 024564  104144                  1$:     ERROR   +144
4166 024566  000435                          BR      XDONE
4167
4168                                ;REPORT FPS ERRORS
4169
4170 024570                        XERR2:
4171 024570  010537  001240                  MOV     R5,$TMP3
4172 024574  010437  001242                  MOV     R4,$TMP4
4173 024600  104145                  1$:     ERROR   +145
4174 024602  000427                          BR      XDONE
4175 024604                        XERR4:
4176 024604  010537  001240                  MOV     R5,$TMP3
4177 024610  010437  001242                  MOV     R4,$TMP4
4178 024614  104146                  1$:     ERROR   +146
4179 024616  000421                          BR      XDONE
4180 024620  000000                XTMP:     .WORD   0
4181 024622  000000                XDAT00:   .WORD   0
4182 024624  000000                XDAT01:   .WORD   0
4183 024626  000000                XDAT02:   .WORD   0
4184 024630  000000                XDAT03:   .WORD   0
4185
4186 024632  010421                XPAT00:   .WORD   010421
4187 024634  021042                XPAT01:   .WORD   021042
4188 024636  031463                XPAT02:   .WORD   031463
4189 024640  042104                XPAT03:   .WORD   042104
4190
4191 024642  000000                XPAT10:   .WORD   0
4192 024644  000000                XAPT11:   .WORD   0
4193 024646  000000                XPAT12:   .WORD   0
4194 024650  000000                XPAT13:   .WORD   0
4195 024652  104210                XPAT20:   .WORD   104210
4196 024654  114631                XPAT21:   .WORD   114631
4197 024656  125252                XPAT22:   .WORD   125252
4198 024660  135673                XPAT23:   .WORD   135673
4199
4200 024662                        XDONE:
     024662  104413                        RSETUP                  ;GO INITIALIZE THE FPS AND STACK; AND
                                                                    ;SEE IF THE USER HAS EXPRESSED
                                                                    ;THE DESIRE TO CHANGE THE SOFTWARE
                                                                    ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
                                                                    ;THE USER TYPED CONTROL G?).
4201
```

```
                                .SBTTL  TEST # 27 - SUBD WITH AC=0 TEST
                                ;;*********************************************************************
                                ;*TEST 27 - SUBD WITH AC=0 TEST
                                ;*
                                ;* THIS IS A TEST OF SUBD WITH AC=0.  BOTH POSITIVE
                                ;* AND NEGATIVE FSRC'S ARE TRIED.
                                ;*
                                ;;*********************************************************************
 024664  000004                TST27:  SCOPE
4210 024666  005037  025222            CLR     YFLAG
4211 024672  012737  025242  025224    MOV     #YPAT00,YTMP1   ;P
4212 024700  012737  025252  025226    MOV     #YPAT10,YTMP2   ;N
4213 024706  012737  000210  025230    MOV     #210,YTMP3
4214 024714                    Y1:
 024714  012737  024722  001110        MOV     #1$,$LPERR      ;SET UP THE LOOP ON ERROR ADDRESS.
4215 024722  012700  000200    1$:     MOV     #200,R0
4216 024726  170100                    LDFPS   R0              ;SET DOUBLE MODE
4217 024730  012700  025262            MOV     #YPAT20,R0      ;SET AC0=0
4218 024734  172410                    LDD     (R0),AC0
4219 024736  013700  025224            MOV     YTMP1,R0
4220 024742  173010            Y2:     SUBD    (R0),AC0        ;TEST INSTRUCTION
4221 024744  170205                    STFPS   R5
4222 024746  170011                    SETD
4223 024750  012700  025232            MOV     #YDAT00,R0      ;GET RESULT
4224 024754  174010                    STD     AC0,(R0)
4225 024756  012702  000004            MOV     #4,R2
4226 024762  013701  025226            MOV     YTMP2,R1        ;CHECK RESULT.
4227 024766  022021            Y3:     CMP     (R0)+,(R1)+
4228 024770  001026                    BNE     Y6
4229 024772  077203                    SOB     R2,Y3
4230 024774  023705  025230            CMP     YTMP3,R5        ;FPS CORRECT?
4231 025000  001401                    BEQ     Y4
4232 025002  000475                    BR      YERR3
4233 025004  005737  025222    Y4:     TST     YFLAG           ;FINISHED TEST?
4234 025010  001015                    BNE     Y5
4235 025012  012737  177777  025222    MOV     #-1,YFLAG
4236 025020  012737  025252  025224    MOV     #YPAT10,YTMP1
4237 025026  012737  025242  025226    MOV     #YPAT00,YTMP2
4238 025034  012737  000200  025230    MOV     #200,YTMP3
4239 025042  000724                    BR      Y1
4240 025044  000512            Y5:     BR      YDONE
4241 025046  012702  000004    Y6:     MOV     #4,R2
4242 025052  012700  025224            MOV     #YTMP1,R0       ;DID XOR OF SIGN BIT
4243 025056  012701  025232            MOV     #YDAT00,R1      ;FAIL?
4244 025062  022021            Y7:     CMP     (R0)+,(R1)+
4245 025064  001002                    BNE     YERR1
4246 025066  077203                    SOB     R2,Y7
4247 025070  000421                    BR      YERR2
4248 025072                    YERR1:                          ;DATA FAILURE
4249 025072  012737  024742  001236    MOV     #Y2,$TMP2
4250 025100  013737  025224  001240    MOV     YTMP1,$TMP3
4251 025106  012737  025262  001242    MOV     #YPAT20,$TMP4
4252 025114  012737  025232  001244    MOV     #YDAT00,$TMP5
4253 025122  013737  025226  001246    MOV     YTMP2,$TMP6
4254 025130  104147            1$:     ERROR   +147
4255 025132  000457                    BR      YDONE
```

```
4256 025134                                YERR2:                          ;XOR OF SIGN BIT
4257 025134  012737  024742  001236                MOV     #Y2,$TMP2       ;FAILED
4258 025142  013737  025224  001240                MOV     YTMP1,$TMP3
4259 025150  012737  025262  001242                MOV     #YPAT20,$TMP4
4260 025156  012737  025232  001244                MOV     #YDAT00,$TMP5
4261 025164  013737  025226  001246                MOV     YTMP2,$TMP6
4262 025172  104150                        1$:     ERROR   +150
4263 025174  000436                                BR      YDONE
4264 025176                                YERR3:                          ;FPS WRONG.
4265 025176  012737  024742  001236                MOV     #Y2,$TMP2
4266 025204  010537  001240                        MOV     R5,$TMP3
4267 025210  013737  025230  001242                MOV     YTMP3,$TMP4
4268 025216  104151                        1$:     ERROR   +151
4269 025220  000424                                BR      YDONE
4270
4271 025222  000000                        YFLAG:  .WORD   0
4272 025224  000000                        YTMP1:  .WORD   0
4273 025226  000000                        YTMP2:  .WORD   0
4274 025230  000000                        YTMP3:  .WORD   0
4275
4276 025232  000000                        YDAT00: .WORD   0
4277 025234  000000                        YDAT01: .WORD   0
4278 025236  000000                        YDAT02: .WORD   0
4279 025240  000000                        YDAT03: .WORD   0
4280
4281 025242  063146                        YPAT00: .WORD   063146
4282 025244  052525                        YPAT01: .WORD   052525
4283 025246  042104                        YPAT02: .WORD   042104
4284 025250  167356                        YPAT03: .WORD   167356
4285
4286 025252  163146                        YPAT10: .WORD   163146
4287 025254  052525                        YPAT11: .WORD   052525
4288 025256  042104                        YPAT12: .WORD   042104
4289 025260  167356                        YPAT13: .WORD   167356
4290
4291 025262  000000                        YPAT20: .WORD   0
4292 025264  000000                        YPAT21: .WORD   0
4293 025266  000000                        YPAT22: .WORD   0
4294 025270  000000                        YPAT23: .WORD   0
4295
4296 025272                                YDONE:
     025272  104413                                RSETUP                  ;GO INITIALIZE THE FPS AND STACK; AND
                                                                            ;SEE IF THE USER HAS EXPRESSED
                                                                            ;THE DESIRE TO CHANGE THE SOFTWARE
                                                                            ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
                                                                            ;THE USER TYPED CONTROL G?).
```

```
                                .SBTTL  TEST # 30 - ADDD WITH AC=0 TEST
                                ;;*****************************************************************
                                ;*TEST 30 - ADDD WITH AC=0 TEST

                                ;@
                                ;@ THIS IS A TEST OF ADDD WITH AC=0.  BOTH
                                ;* POSITIVE AND NEGATIVE FSRC'S ARE TRIED.
                                ;*

                                ;;*****************************************************************
     025274  000004             TST30:  SCOPE
4306 025276  005037  025532             CLR     ZFLAG
4307 025302  012737  025550  025534     MOV     #ZPAT00,ZTMP1   ;P
4308 025310  012737  000200  025536     MOV     #200,ZTMP2
4309 025316                     Z1:
     025316  012737  025324  001110     MOV     #1$,$LPERR      ;SET UP THE LOOP ON ERROR ADDRESS.
4310 025324  012700  000200     1$:     MOV     #200,R0
4311 025330  170100                     LDFPS   R0              ;SET DOUBLE MODE
4312 025332  012700  025570             MOV     #ZPAT20,R0      ;SET AC0=0
4313 025336  172410                     LDD     (R0),AC0
4314 025340  013700  025534             MOV     ZTMP1,R0
4315 025344  172010             Z2:     ADDD    (R0),AC0        ;TEST INSTRUCTION
4316 025346  170205                     STFPS   R5
4317 025350  170011                     SETD
4318 025352  012700  025540             MOV     #ZDAT00,R0      ;GET RESULT
4319 025356  174010                     STD     AC0,(R0)
4320 025360  012702  000004             MOV     #4,R2
4321 025364  013701  025534             MOV     ZTMP1,R1        ;RESULT CORRECT?
4322 025370  022021             Z3:     CMP     (R0)+,(R1)+
4323 025372  001401                     BEQ     Z4
4324 025374  000423                     BR      ZERR1
4325 025376  077204             Z4:     SOB     R2,Z3
4326 025400  023705  025536             CMP     ZTMP2,R5        ;FPS CORRECT?
4327 025404  001401                     BEQ     Z5
4328 025406  000437                     BR      ZERR2
4329 025410  005737  025532     Z5:     TST     ZFLAG           ;FINISHED TEST?
4330 025414  001012                     BNE     Z6
4331 025416  012737  177777  025532     MOV     #-1,ZFLAG
4332 025424  012737  025560  025534     MOV     #ZPAT10,ZTMP1
4333 025432  012737  000210  025536     MOV     #210,ZTMP2
4334 025440  000726                     BR      Z1
4335 025442  000456             Z6:     BR      ZDONE
4336 025444                     ZERR1:                          ;DATA FAILURE
4337 025444  012737  025344  001236             MOV     #Z2,$TMP2
4338 025452  013737  025534  001240             MOV     ZTMP1,$TMP3
4339 025460  012737  025570  001242             MOV     #ZPAT20,$TMP4
4340 025466  012737  025540  001244             MOV     #ZDAT00,$TMP5
4341 025474  013737  025534  001246             MOV     ZTMP1,$TMP6
4342 025502  104152             1$:     ERROR   +152
4343 025504  000435                     BR      ZDONE
4344 025506                     ZERR2:
4345 025506  012737  025344  001236             MOV     #Z2,$TMP2
4346 025514  010537  001240                     MOV     R5,$TMP3
4347 025520  013737  025536  001242             MOV     ZTMP2,$TMP4
4348 025526  104153             1$:     ERROR   +153
4349 025530  000423                     BR      ZDONE
4350
```

```
4351 025532  000000                ZFLAG:  .WORD   0
4352 025534  000000                ZTMP1:  .WORD   0
4353 025536  000000                ZTMP2:  .WORD   0
4354
4355 025540  000000                ZDAT00: .WORD   0
4356 025542  000000                ZDAT01: .WORD   0
4357 025544  000000                ZDAT02: .WORD   0
4358 025546  000000                ZDAT03: .WORD   0
4359
4360 025550  031463                ZPAT00: .WORD   031463
4361 025552  010421                ZPAT01: .WORD   010421
4362 025554  146314                ZPAT02: .WORD   146314
4363 025556  156735                ZPAT03: .WORD   156735
4364
4365 025560  156735                ZPAT10: .WORD   156735
4366 025562  167356                ZPAT11: .WORD   167356
4367 025564  135673                ZPAT12: .WORD   135673
4368 025566  146314                ZPAT13: .WORD   146314
4369
4370 025570  000000                ZPAT20: .WORD   0
4371 025572  000000                ZPAT21: .WORD   0
4372 025574  000000                ZPAT22: .WORD   0
4373 025576  000000                ZPAT23: .WORD   0
4374
4375 025600                        7DONE:
     025600  104413                        RSETUP                  ;GO INITIALIZE THE FPS AND STACK; AND
                                                                    ;SEE IF THE USER HAS EXPRESSED
                                                                    ;THE DESIRE TO CHANGE THE SOFTWARE
                                                                    ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
                                                                    ;THE USER TYPED CONTROL G?).
4376
4377
```

```
4385                                    .SBTTL  TEST # 31 - ADDF & ADDD E(AC)=E(FSRC) & (BUT FT) TEST
                                        ;;*****************************************************************
                                        ;*TEST 31 - ADDF & ADDD E(AC)=E(FSRC) & (BUT FT) TEST
                                        ;*
                                        ;* THIS IS A TEST OF THE ADD INSTRUCTION WITH THE
                                        ;* OPERANDS HAVING EQUAL EXPONENTS.  THE (BUF FT)
                                        ;* FORK IN THE ROUND/TRUNK FLOWS IS ALSO TESTED.
                                        ;*
                                        ;;*****************************************************************
     025602  000004                     TST31:  SCOPE
4386 025604
     025604  012737  025612  001110             AA1:    MOV     #1$,$LPERR      ;SET UP THE LOOP ON ERROR ADDRESS.
4387 025612  012700  003240             1$:     MOV     #3240,R0
4388 025616  170100                             LDFPS   R0              ;SET FIV FIV FD AND FT
4389 025620  012737  026200  000244             MOV     #AAERR0,FPVECT  ;IN CASE THE OVER/UNDER
4390 025626  012700  026556                     MOV     #AAPAT0,R0      ;FLOWS IN TRAP WILL
4391                                                                    ;OCCUR
4392 025632  172410                             LDD     (R0),AC0        ;SET UP AC0
4393 025634  012737  025646  001236             MOV     #AA2,$TMP2      ;OPERAND
4394 025642  012700  026566                     MOV     #AAPAT1,R0
4395 025646  172010                     AA2:    ADDD    (R0),AC0        ;TEST INSTRUCTION
4396                                                                    ;SHOULD TRUNCATE
4397 025650  012700  026546             AA3:    MOV     #AADAT0,R0
4398 025654  174010                             STD     AC0,(R0)        ;GET THE RESULT
4399 025656  012701  026576                     MOV     #AAPAT2,R1
4400 025662  012702  000004                     MOV     #4,R2
4401 025666  022021                     AA4:    CMP     (R0)+,(R1)+     ;CORRECT?
4402 025670  001414                             BEQ     AA7
4403 025672  012700  026606                     MOV     #AAPAT3,R0      ;DID (BUT FT) FAIL
4404 025676  012701  026546                     MOV     #AADAT0,R1
4405 025702  012702  000004                     MOV     #4,R2
4406 025706  022021                     AA5:    CMP     (R0)+,(R1)+
4407 025710  001401                             BEQ     AA6
4408 025712  000565                             BR      AAERR1          ;DATA ERROR
4409 025714  077204                     AA6:    SOB     R2,AA5
4410 025716  000137  026322                     JMP     AAERR2          ;(BUT FT) ERROR
4411 025722  077217                     AA7:    SOB     R2,AA4
4412
4413                                    ;NOW TEST DOUBLE FLOATING ROUND MODE.
4414
4415 025724                             AA10:
     025724  012737  025732  001110             MOV     #1$,$LPERR      ;SET UP THE LOOP ON ERROR ADDRESS.
4416 025732  012700  003200             1$:     MOV     #3200,R0        ;SET FD FIV FIV. FT=0
4417 025736  170100                             LDFPS   R0
4418 025740  012700  026556                     MOV     #AAPAT0,R0
4419 025744  172410                             LDD     (R0),AC0        ;SET UP AC0 OPERAND
4420 025746  012737  025760  001236             MOV     #AA11,$TMP2
4421 025754  012700  026566                     MOV     #AAPAT1,R0
4422 025760  172010                     AA11:   ADDD    (R0),AC0        ;TEST INSTRUCTION
4423                                                                    ;SHOULD ROUND
4424 025762  012700  026546             AA12:   MOV     #AADAT0,R0
4425 025766  174010                             STD     AC0,(R0)        ;GET THE RESULT
4426 025770  012701  026606                     MOV     #AAPAT3,R1
4427 025774  012702  000004                     MOV     #4,R2
4428 026000  022021                     AA13:   CMP     (R0)+,(R1)+     ;CORRECT?
4429 026002  001425                             BEQ     AA20
4430 026004  012700  026576                     MOV     #AAPAT2,R0      ;DID (BUT FT) FAIL?
```

```
                                                MOV     #AADATO,R1
                                                MOV     #4,R2
                                        AA14:   CMP     (R0)+,(R1)+
                                                BEQ     AA17
                                                MOV     #AAPAT4,R0      ;WAS THE FLOATING
                                                MOV     #AADATO,R1      ;CONSTANT USED
                                                MOV     #4,R2           ;INSTEAD OF THE
                                        AA15:   CMP     (R0)+,(R1)+     ;DOUBLE CONSTANT
                                                BEQ     AA16            ;IN THE ROUND
                                                BR      AAERR3  ;FLOWS?
                                        AA16:   SOB     R2,AA15         ;DATA ERROR
                                                BR      AAERR4  ;CONSTANT ERROR
                                        AA17:   SOB     R2,AA14
                                                BR      AAERR5  ;(BUT FT) ERROR
                                        AA20:   SOB     R2,AA13

                                        ;NOW TEST ADDF WITH FT=0, ROUND MODE

                                        AA21:
                                        1$:     MOV     #1$,$LPERR      ;SET UP THE LOOP ON ERROR ADDRESS.
                                                MOV     #3200,R0                ;FIV=1, FIV=1, FT=0
                                                LDFPS   R0
                                                MOV     #AAPAT0,R0      ;LOAD AC0 OPERAND
                                                LDD     (R0),AC0
                                                SETF                    ;ENTER FLOATING MODE
                                                MOV     #AA22,$TMP2
                                                MOV     #AAPAT5,R0
                                        AA22:   ADDF    (R0),AC0        ;TEST INSTRUCTION
                                                                        ;SHOULD ROUND
                                        AA23:
                                                SETD                    ;RESET TO DOUBLE
                                                                        ;MODE
                                                MOV     #AADATO,R0      ;GET THE RESULT
                                                STD     AC0,(R0)
                                                MOV     #AAPAT6,R1      ;CORRECT?
                                                MOV     #2,R2
                                        AA24:   CMP     (R0)+,(R1)+
                                                BEQ     AA27
                                                MOV     #AAPAT2,R0      ;WAS THE DOUBLE
                                                MOV     #AADATO,R1      ;CONSTANT USED INSTEAD
                                                MOV     #2,R2           ;OF THE FLOATING
                                        AA25:   CMP     (R0)+,(R1)      ;CONSTANT IN THE
                                                BEQ     AA26            ;ROUND FLOWS?
                                                BR      AAERR6          ;DATA ERROR
                                        AA26:   SOB     R2,AA25
                                                BR      AAERR7          ;CONSTANT ERROR
                                        AA27:   SOB     R2,AA24
                                                JMP     AADONE

                                        ;COME HERE IF A TRAP OCCURS TO 244.

                                        AAERR0: MOV     $TMP2,R0        ;SEE IF THE TRAP WAS
                                                TST     (R0)+           ;AT A TEST INSTRUCTION
                                                CMP     R0,(SP)
                                                BEQ     1$
                                        10$:    JMP     FPSPUR
                                        1$:
```

```
                                                        STST    R0              ;GET FEC
                                                        CMP     R0,#10
                                                        BEQ     20$             ;OVERFLOW
                                                        CMP     R0,#12
                                                        BEQ     30$             ;UNDERFLOW
                                                        BR      10$
                                                20$
                                                20$:    MOV     (SP),$TMP2      ;REPORT OVERFLOW ERROR
                                                        CMP     (SP)+,(SP)+
                                                21$:    ERROR   +154
                                                25$:    JMP     AADONE
                                                30$:    MOV     (SP),$TMP2      ;REPORT UNDERFLOW
                                                        CMP     (SP)+,(SP)+     ;ERROR
                                                31$:    ERROR   +155
                                                        BR      25$

                                                ;ADDD RESULT INCORRECT
                                                AAERR1: MOV     #AAPAT2,$TMP6
                                                AAER10: MOV     #AAPAT0,$TMP4
                                                        MOV     #AAPAT1,$TMP3
                                                        MOV     #AADAT0,$TMP5
                                                1$:     ERROR   +162
                                                        BR      AADONE
                                                AAERR2: MOV     #AAPAT2,$TMP6   ;(BUT FT) FAILED.
                                                        MOV     #AAPAT0,$TMP4
                                                        MOV     #AAPAT1,$TMP3
                                                        MOV     #AADAT0,$TMP5
                                                1$:     ERROR   +156
                                                        BR      AADONE
                                                AAERR3: MOV     #AAPAT3,$TMP6   ;DATA ERROR.
                                                        BR      AAER10
                                                AAERR4: MOV     #AAPAT3,$TMP6   ;BAD CONSTANT
                                                        MOV     #AAPAT0,$TMP4
                                                        MOV     #AAPAT1,$TMP3
                                                        MOV     #AADAT0,$TMP5
                                                1$:     ERROR   +160
                                                        BR      AADONE
                                                AAERR5: MOV     #AAPAT3,$TMP6   ;(BUT FT) FAILED.
                                                        MOV     #AAPAT0,$TMP4
                                                        MOV     #AAPAT1,$TMP3
                                                        MOV     #AADAT0,$TMP5
                                                1$:     ERROR   +157
                                                        BR      AADONE
                                                AAERR6: MOV     #AAPAT5,$TMP3   ;FD=0 AND
                                                        MOV     #AAPAT0,$TMP4   ;DATA ERROR
                                                        MOV     #AADAT0,$TMP5
                                                        MOV     #AAPAT6,$TMP6
                                                1$:     ERROR   +160
                                                        BR      AADONE
                                                AAERR7: MOV     #AAPAT5,$TMP3   ;CONSTANT ERROR
                                                        MOV     #AAPAT0,$TMP4
                                                        MOV     #AADAT0,$TMP5
                                                        MOV     #AAPAT6,$TMP6
                                                1$:     ERROR   +161
                                                        BR      AADONE
                                                AADAT0: .WORD   0,0,0,0
                                                AAPAT0: .WORD   200,0,0,0
```

```
 4544 026566  000200  000000  000000  AAPAT1: .WORD   200,0,0,1
 4545 026576  000400  000000  000000  AAPAT2: .WORD   400,0,0,0
 4546 026606  000400  000000  000000  AAPAT3: .WORD   400,0,0,1
 4547 026616  000400  000000  100000  AAPAT4: .WORD   400,0,100000,0
 4548 026626  000200  000001  000000  AAPAT5: .WORD   200,1,0,0
 4549 026636  000400  000001  000000  AAPAT6: .WORD   400,1,0,0
 4550 026646                          AADONE:
      026646  104413                          RSETUP                  ;GO INITIALIZE THE FPS AND STACK; AND
                                                                      ;SEE IF THE USER HAS EXPRESSED
                                                                      ;THE DESIRE TO CHANGE THE SOFTWARE
                                                                      ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
                                                                      ;THE USER TYPED CONTROL G?).
```

```
4561                                    .SBTTL  TEST # 32 - ADDF & ADDD WITH E(AC) LESS THAN E(FSRC) TEST
                                        ;;*****************************************************************
                                        ;*TEST 32 - ADDF & ADDD WITH E(AC) LESS THAN E(FSRC) TEST
                                        ;*
                                        ;*THIS IS ATEST OF THE ADDD AND ADDF
                                        ;*INSTRUCTIONS AND THE ALIGN AC ALGORITHM
                                        ;*FLOWS.  THE CONSTANT (25 FOR FLOATING, 57 FOR
                                        ;*DOUBLE) USED IS CHECKED.  THEN SIMPLE
                                        ;*AND WORST CASE ALIGNMENT SITUATIONS ARE
                                        ;*TRIED.  NOTE E(AC) IS LESS THEN E(FSRC)
                                        ;*
                                        ;;*****************************************************************
    026650  000004                      TST32:  SCOPE
4562                                                                ;EXPONENT DIFFERENCE=57=71 (OCT) FD=1
4563 026652                             CC1:
    026652  012737  026660  001110              MOV     #1$,$LPERR      ;SET UP THE LOOP ON ERROR ADDRESS.
4564 026660  012704  003200             1$:     MOV     #3200,R4        ;SET FIV,FIV, AND FD
4565 026664  170104                             LDFPS   R4
4566 026666  012737  026706  001236             MOV     #CC2,$TMP2
4567 026674  012700  030334                     MOV     #CCP0,R0        ;SET AC0 OPERAND
4568 026700  172410                             LDD     (R0),AC0                ;AC0
4569 026702  012700  030354                     MOV     #CCP2,R0
4570 026706  172010                     CC2:    ADDD    (R0),AC0        ;TEST INSTRUCTION
4571 026710  170205                             STFPS   R5              ;GET FPS
4572 026712  012700  030324                     MOV     #CCDAT0,R0      ;GET THE RESULT
4573 026716  174010                             STD     AC0,(R0)
4574 026720  012701  030354                     MOV     #CCP2,R1        ;IS IT CORRECT
4575 026724  012702  000004                     MOV     #4,R2
4576 026730  022021                     CC3:    CMP     (R0)+,(R1)+
4577 026732  001415                             BEQ     CC6
4578 026734  012700  030324                     MOV     #CCDAT0,R0      ;DID A BAD
4579 026740  012701  030334                     MOV     #CCP0,R1        ;CONSTANT (NOT 57)
4580 026744  012702  000004                     MOV     #4,R2           ;GET GENERATED
4581 026750  022021                     CC4:    CMP     (R0)+,(R1)+     ;FOR THE ALIGNMENT
4582 026752  001402                             BEQ     CC5             ;FLOWS?
4583 026754  000137  027722                     JMP     CCER1           ;DATA ERROR.D
4584 026760  077205                     CC5:    SOB     R2,CC4
4585 026762  000137  027760                     JMP     CCER2           ;BAD CONSTANT.D
4586 026766  077220                     CC6:    SOB     R2,CC3
4587 026770  020405                             CMP     R4,R5           ;FPS CORRECT?
4588 026772  001402                             BEQ     CC7
4589 026774  000137  027666                     JMP     CCER0           ;BAD FPS.
4590                                    ;EXPONENT DIFFERENCE=56=70 (OCT) FD=1
4591 027000                             CC7:
    027000  012737  027006  001110              MOV     #1$,$LPERR      ;SET UP THE LOOP ON ERROR ADDRESS.
4592 027006  012704  003200             1$:     MOV     #3200,R4        ;SET FIV,FIV, AND FD
4593 027012  170104                             LDFPS   R4
4594 027014  012737  027034  001236             MOV     #CC8,$TMP2
4595 027022  012700  030334                     MOV     #CCP0,R0        ;SET AC0 OPERAND
4596 027026  172410                             LDD     (R0),AC0
4597 027030  012700  030344                     MOV     #CCP1,R0        ;FSRC
4598 027034  172010                     CC8:    ADDD    (R0),AC0        ;TEST INSTRUCTION
4599 027036  170205                             STFPS   R5              ;GET FPS
4600 027040  012700  030324                     MOV     #CCDAT0,R0      ;GET THE RESULT
4601 027044  174010                             STD     AC0,(R0)
4602 027046  012701  030424                     MOV     #CCP7,R1        ;IS IT CORRECT
4603 027052  012702  000004                     MOV     #4,R2
```

```
 4604 027056  022021                    CC9:    CMP     (R0)+,(R1)+
 4605 027060  001415                            BEQ     CC12
 4606 027062  012700  030324                    MOV     #CCDAT0,R0      ;DID A BAD
 4607 027066  012701  030344                    MOV     #CCP1,R1        ;CONSTANT (NOT 57)
 4608 027072  012702  000004                    MOV     #4,R2           ;GET GENERATED
 4609 027076  022021                    CC10:   CMP     (R0)+,(R1)+     ;FOR THE ALIGNMENT
 4610 027100  001402                            BEQ     CC11            ;FLOWS?
 4611 027102  000137  030016                    JMP     CCER3           ;DATA ERROR.D
 4612 027106  077205                    CC11:   SOB     R2,CC10
 4613 027110  000137  030034                    JMP     CCER4           ;BAD CONSTANT.D
 4614 027114  077220                    CC12:   SOB     R2,CC9
 4615 027116  020405                            CMP     R4,R5           ;FPS CORRECT?
 4616 027120  001402                            BEQ     CC13
 4617 027122  000137  027666                    JMP     CCER0           ;BAD FPS.
 4618                                   ;EXPONENT DIFFERENCE=25=31 (OCT) FD=0
 4619 027126                            CC13:
      027126  012737  027134  001110            MOV     #1$,$LPERR      ;SET UP THE LOOP ON ERROR ADDRESS.
 4620 027134  012737  027162  001236  1$:     MOV     #CC14,$TMP2
 4621 027142  012700  030334                    MOV     #CCP0,R0        ;SET UP AC0 OPERAND.
 4622 027146  172410                            LDD     (R0),AC0
 4623 027150  012704  003000                    MOV     #3000,R4        ;SET FIV,FIV. CLEAR FD.
 4624 027154  170104                            LDFPS   R4
 4625 027156  012700  030414                    MOV     #CCP6,R0        ;FSRC
 4626 027162  172010                    CC14:   ADDF    (R0),AC0        ;TEST INSTRUCTION
 4627 027164  170205                            STFPS   R5
 4628 027166  170011                            SETD                    ;REENTER DOUBLE MOVE
 4629 027170  012700  030324                    MOV     #CCDAT0,R0      ;GET THE RESULT
 4630 027174  174010                            STD     AC0,(R0)
 4631 027176  012701  030414                    MOV     #CCP6,R1        ;IS THE RESULT CORRECT?
 4632 027202  012702  000002                    MOV     #2,R2
 4633 027206  022021                    CC15:   CMP     (R0)+,(R1)+
 4634 027210  001415                            BEQ     CC18
 4635 027212  012700  030324                    MOV     #CCDAT0,R0      ;WAS A BAD CONSTANT
 4636 027216  012701  030364                    MOV     #CCP3,R1        ;USED (NOT 25) IN
 4637 027222  012702  000002                    MOV     #2,R2           ;THE ALIGN FLOWS?
 4638 027226  022021                    CC16:   CMP     (R0)+,(R1)+
 4639 027230  001402                            BEQ     CC17
 4640 027232  000137  030072                    JMP     CCER5           ;DATA ERROR F
 4641 027236  077205                    CC17:   SOB     R2,CC16
 4642 027240  000137  030126                    JMP     CCER6           ;BAD CONSTANT F
 4643 027244  077220                    CC18:   SOB     R2,CC15
 4644 027246  020405                            CMP     R4,R5
 4645 027250  001402                            BEQ     CC19
 4646 027252  000137  027704                    JMP     CCER90          ;BAD FPS.
 4647                                   ;EXPONENT DIFFERENCE=24=30 (OCT) FD=0
 4648 027256                            CC19:
      027256  012737  027264  001110            MOV     #1$,$LPERR      ;SET UP THE LOOP ON ERROR ADDRESS.
 4649 027264  012737  027312  001236  1$:     MOV     #CC20,$TMP2
 4650 027272  012700  030364                    MOV     #CCP3,R0        ;SET UP AC0 OPERAND.
 4651 027276  172410                            LDD     (R0),AC0
 4652 027300  012704  003000                    MOV     #3000,R4        ;SET FIV,FIV. CLEAR FD.
 4653 027304  170104                            LDFPS   R4
 4654 027306  012700  030404                    MOV     #CCP5,R0        ;FSRC
 4655 027312  172010                    CC20:   ADDF    (R0),AC0        ;TEST INSTRUCTION
 4656 027314  170205                            STFPS   R5
 4657 027316  170011                            SETD                    ;REENTER DOUBLE MOVE
 4658 027320  012700  030324                    MOV     #CCDAT0,R0      ;GET THE RESULT
```

```
4659 027324  174010                        STD     AC0,(R0)
4660 027326  012701  030434                MOV     #CCP10,R1       ;IS THE RESLT CORRECT?
4661 027332  012702  000002                MOV     #2,R2
4662 027336  022021                CC21:   CMP     (R0)+,(R1)+
4663 027340  001415                        BEQ     CC24
4664 027342  012700  030324                MOV     #CCDATO,R0      ;WAS A BAD CONSTANT
4665 027346  012701  030404                MOV     #CCP5,R1        ;USED (NOT 25) IN
4666 027352  012702  000002                MOV     #2,R2           ;THE ALIGN FLOWS?
4667 027356  022021                CC22:   CMP     (R0)+,(R1)+
4668 027360  001402                        BEQ     CC23
4669 027362  000137  030162                JMP     CCER7           ;DATA ERROR F
4670 027366  077205                CC23:   SOB     R2,CC22
4671 027370  000137  030200                JMP     CCER8           ;BAD CONSTANT F
4672 027374  077220                CC24:   SOB     R2,CC21
4673 027376  020405                        CMP     R4,R5
4674 027400  001402                        BEQ     CC25
4675 027402  000137  027704                JMP     CCER90          ;BAD FPS.
4676                               ;EXPONENT DIFFERENCE=1 FD=1
4677 027406                        CC25:
     027406  012737  027414  001110        MOV     #1$,$LPERR      ;SET UP THE LOOP ON ERROR ADDRESS.
4678 027414  012704  003200        1$:     MOV     #3200,R4        ;SET FIV,FIV, AND FD
4679 027420  170104                        LDFPS   R4
4680 027422  012737  027442  001236        MOV     #CC26,$TMP2
4681 027430  012700  030334                MOV     #CCP0,R0        ;SET AC0 OPERAND
4682 027434  172410                        LDD     (R0),AC0
4683 027436  012700  030364                MOV     #CCP3,R0        ;FSRC
4684 027442  172010                CC26:   ADDD    (R0),AC0        ;TEST INSTRUCTION
4685 027444  170205                        STFPS   R5              ;GET FPS
4686 027446  012700  030324                MOV     #CCDATO,R0      ;GET THE RESULT
4687 027452  174010                        STD     AC0,(R0)
4688 027454  012701  030444                MOV     #CCP11,R1       ;IS IT CORRECT
4689 027460  012702  000004                MOV     #4,R2
4690 027464  022021                CC27:   CMP     (R0)+,(R1)+
4691 027466  001415                        BEQ     CC30
4692 027470  012700  030324                MOV     #CCDATO,R0      ;DID A BAD
4693 027474  012701  030364                MOV     #CCP3,R1        ;CONSTANT (NOT 57)
4694 027500  012702  000004                MOV     #4,R2           ;GET GENERATED
4695 027504  022021                CC28:   CMP     (R0)+,(R1)+     ;FOR THE ALIGNMENT
4696 027506  001402                        BEQ     CC29            ;FLOWS?
4697 027510  000137  030234                JMP     CCER10          ;DATA ERROR.D
4698 027514  077205                CC29:   SOB     R2,CC28
4699 027516  000137  030252                JMP     CCER11          ;BAD CONSTANT.D
4700 027522  077220                CC30:   SOB     R2,CC27
4701 027524  020405                        CMP     R4,R5           ;FPS CORRECT?
4702 027526  001402                        BEQ     CC31
4703 027530  000137  027666                JMP     CCER0           ;BAD FPS.
4704                               ;EXPONENT DIFFERENCE=100=144 (OCT) FD=1
4705 027534                        CC31:
     027534  012737  027542  001110        MOV     #1$,$LPERR      ;SET UP THE LOOP ON ERROR ADDRESS.
4706 027542  012704  003200        1$:     MOV     #3200,R4        ;SET FIV,FIV, AND FD
4707 027546  170104                        LDFPS   R4
4708 027550  012737  027570  001236        MOV     #CC32,$TMP2
4709 027556  012700  030334                MOV     #CCP0,R0        ;SET AC0 OPERAND
4710 027562  172410                        LDD     (R0),AC0
4711 027564  012700  030374                MOV     #CCP4,R0        ;FSRC
4712 027570  172010                CC32:   ADDD    (R0),AC0        ;TEST INSTRUCTION
4713 027572  170205                        STFPS   R5              ;GET FPS
```

TEST # 32 - ADDF & ADDD WITH E(AC) LESS THAN E(FSRC) TEST

```
4714 027574  012700  030324                    MOV     #CCDATO,R0      ;GET THE RESULT
4715 027600  174010                            STD     AC0,(R0)
4716 027602  012701  030374                    MOV     #CCP4,R1        ;IS IT CORRECT
4717 027606  012702  000004                    MOV     #4,R2
4718 027612  022021                CC33:       CMP     (R0)+,(R1)+
4719 027614  001415                            BEQ     CC36
4720 027616  012700  030324                    MOV     #CCDATO,R0      ;DID A BAD
4721 027622  012701  030374                    MOV     #CCP4,R1        ;CONSTANT (NOT 57)
4722 027626  012702  000004                    MOV     #4,R2           ;GET GENERATED
4723 027632  022021                CC34:       CMP     (R0)+,(R1)+     ;FOR THE ALIGNMENT
4724 027634  001402                            BEQ     CC35            ;FLOWS?
4725 027636  000137  030270                    JMP     CCER12          ;DATA ERROR.D
4726 027642  077205                CC35:       SOB     R2,CC34
4727 027644  000137  030306                    JMP     CCER13          ;BAD CONSTANT.D
4728 027650  077220                CC36:       SOB     R2,CC33
4729 027652  020405                            CMP     R4,R5           ;FPS CORRECT?
4730 027654  001402                            BEQ     CC37
4731 027656  000137  027666                    JMP     CCER0           ;BAD FPS.
4732 027662  000137  030464        CC37:       JMP     CCDONE
4733 027666  010437  001242        CCER0:      MOV     R4,$TMP4        ;FPS ERROR D
4734 027672  010537  001240                    MOV     R5,$TMP3
4735 027676  104164                1$:         ERROR   +164
4736 027700  000137  030464                    JMP     CCDONE
4737 027704  010437  001242        CCER90:     MOV     R4,$TMP4        ;FPS ERROR F
4738 027710  010537  001240                    MOV     R5,$TMP3
4739 027714  104165                1$:         ERROR   +165
4740 027716  000137  030464                    JMP     CCDONE
4741 027722  012737  030354  001240  CCER1:    MOV     #CCP2,$TMP3     ;DATA ERROR D
4742 027730  012737  030354  001246            MOV     #CCP2,$TMP6
4743 027736  012737  030334  001242  CCER50:   MOV     #CCP0,$TMP4
4744 027744  012737  030324  001244            MOV     #CCDATO,$TMP5
4745 027752  104166                1$:         ERROR   +166
4746 027754  000137  030464                    JMP     CCDONE
4747 027760  012737  030354  001240  CCER2:    MOV     #CCP2,$TMP3     ;CONSTANT BAD D(B)
4748 027766  012737  030354  001246            MOV     #CCP2,$TMP6
4749 027774  012737  030334  001242  CCER22:   MOV     #CCP0,$TMP4
4750 030002  012737  030324  001244            MOV     #CCDATO,$TMP5
4751 030010  104172                1$:         ERROR   +172
4752 030012  000137  030464                    JMP     CCDONE
4753 030016  012737  030344  001240  CCER3:    MOV     #CCP1,$TMP3
4754 030024  012737  030424  001246            MOV     #CCP7,$TMP6
4755 030032  000741                            BR      CCER50
4756 030034  012737  030344  001240  CCER4:    MOV     #CCP1,$TMP3     ;CONSTANT BAD D(G)
4757 030042  012737  030424  001246            MOV     #CCP7,$TMP6
4758 030050  012737  030334  001242  CCER44:   MOV     #CCP0,$TMP4
4759 030056  012737  030324  001244            MOV     #CCDATO,$TMP5
4760 030064  104173                1$:         ERROR   +173
4761 030066  000137  030464                    JMP     CCDONE
4762 030072  012737  030414  001240  CCER5:    MOV     #CCP6,$TMP3     ;DATA ERROR F
4763 030100  012737  030414  001246            MOV     #CCP6,$TMP6
4764 030106  012737  030334  001242  CCER55:   MOV     #CCP0,$TMP4
4765 030114  012737  030324  001244            MOV     #CCDATO,$TMP5
4766 030122  104170                1$:         ERROR   +170
4767 030124  000557                            BR      CCDONE
4768 030126  012737  030414  001240  CCER6:    MOV     #CCP6,$TMP3     ;CONSTANT BAD F(B)
4769 030134  012737  030414  001246            MOV     #CCP6,$TMP6
4770 030142  012737  030334  001242            MOV     #CCP0,$TMP4
```

```
                                                MOV     #CCDATO,$TMP5
                                        1$:     ERROR   +174
                                                BR      CCDONE
                                CCER7:  MOV     #CCP5,$TMP3     ;DATA ERROR F
                                                MOV     #CCP10,$TMP6
                                                BR      CCER55
                                CCER8:  MOV     #CCP5,$TMP3     ;CONSTANT BAD F(G)
                                                MOV     #CCP10,$TMP6
                                                MOV     #CCDATO,$TMP5
                                                MOV     #CCP0,$TMP4
                                        1$:     ERROR   +175
                                                BR      CCDONE
                                CCER10: MOV     #CCP3,$TMP3     ;DATA ERROR D
                                                MOV     #CCP11,$TMP6
                                                BR      CCER50
                                CCER11: MOV     #CCP3,$TMP3     ;CONSTANT BAD D(G)
                                                MOV     #CCP11,$TMP6
                                                BR      CCER44
                                CCER12: MOV     #CCP4,$TMP3     ;DATA ERROR D
                                                MOV     #CCP4,$TMP6
                                                BR      CCER50
                                CCER13: MOV     #CCP4,$TMP3     ;CONSTANT BAD D(B)
                                                MOV     #CCP4,$TMP6
                                                BR      CCER22
000000  000000  000000  CCDATO: .WORD   0,0,0,0
000200  000000  000000  CCP0:   .WORD   200,0,0,0       ;E(AC)=1
016200  000000  000000  CCP1:   .WORD   16200,0,0,0     ;E(FSRC)=E(AC)+56=57
016400  000000  000000  CCP2:   .WORD   16400,0,0,0     ;E(FSRC)=E(AC)+57=58
000400  000000  000000  CCP3:   .WORD   400,0,0,0       ;E(FSRC)=E(AC)+1=2
031200  000000  000000  CCP4:   .WORD   31200,0,0,0     ;E(FSRC)=E(AC)+100=101=145(OCT)
006200  000000  000000  CCP5:   .WORD   6200,0,0,0      ;E(FSRC)=E(AC)+24=25=31(OCT)
006400  000000  000000  CCP6:   .WORD   6400,0,0,0      ;E(FSRC)=E(AC)+25=26=32(OCT)
016200  000000  000000  CCP7:   .WORD   16200,0,0,1     ;CCP1 RES
006200  000001  000000  CCP10:  .WORD   6200,1,0,0      ;CCP5 RES
000500  000000  000000  CCP11:  .WORD   500,0,0,0       ;CCP3 RES
000200  000000  000000  CCP12:  .WORD   200,0,0,0       ;BAD CONSTANT ;RES CCP2,CCP4
                        CCDONE:
104413                  RSETUP                          ;GO INITIALIZE THE FPS AND STACK; AND
                                                        ;SEE IF THE USER HAS EXPRESSED
                                                        ;THE DESIRE TO CHANGE THE SOFTWARE
                                                        ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
                                                        ;THE USER TYPED CONTROL G?).

4808
```

```
 4819                                   .SBTTL  TEST # 33 - ADDF & ADDD WITH E(AC) GREATER THAN E(FSRC) TEST
                                        ;;**********************************************************************
                                        ;*TEST 33 - ADDF & ADDD WITH E(AC) GREATER THAN E(FSRC) TEST
                                        ;*
                                        ;*THIS IS A TEST OF THE ADDD AND ADDF
                                        ;*INSTRUCTIONS AND THE ALIGN FSRC ALGORITHM
                                        ;*FLOWS.  FIRST THE CONSTANT USED IS CHECKED.
                                        ;*THEN SIMPLE AND WORST CASE ALIGNMENT
                                        ;*SITUATIONS ARE TRIED.  NOTE E(AC)
                                        ;*IS GREATER THAN E(FSRC).
                                        ;*
                                        ;;**********************************************************************
     030466  000004                     TST33:  SCOPE
 4820                                   ;EXPONENT DIFFERENCE=57=71 (OCT) FD=1
 4821 030470                            BB1:
     030470  012737  030476  001110             MOV     #1$,$LPERR      ;SET UP THE LOOP ON ERROR ADDRESS.
 4822 030476  012704  003200             1$:    MOV     #3200,R4        ;SET FIV FIV, AND FD
 4823 030502  170104                            LDFPS   R4
 4824 030504  012737  031356  000244             MOV     #BBER0,FPVECT   ;SET UP FOR ERROR
 4825 030512  012737  030532  001236             MOV     #BB2,$TMP2      ;IN CASE THE OVER\
 4826                                                                    ;UNDER FLOWS FAIL.
 4827 030520  012700  031720                    MOV     #BBPAT2,R0      ;SET AC0 OPERAND.
 4828 030524  172410                            LDD     (R0),AC0
 4829 030526  012700  031710                    MOV     #BBPAT1,R0      ;FSRC
 4830 030532  172010                    BB2:    ADDD    (R0),AC0        ;TEST INSTRUCTION
 4831 030534  170205                            STFPS   R5
 4832 030536  012700  031670            BB3:    MOV     #BBDAT0,R0      ;GET THE RESULT
 4833 030542  174010                            STD     AC0,(R0)
 4834 030544  012701  031720                    MOV     #BBPAT2,R1      ;RESULT CORRECT?
 4835 030550  012702  000004                    MOV     #4,R2
 4836 030554  022021                    BB4:    CMP     (R0)+,(R1)+
 4837 030556  001402                            BEQ     BB5
 4838 030560  000137  031416                    JMP     BBER1           ;DATA ERROR D
 4839 030564  077205                    BB5:    SOB     R2,BB4
 4840                                                                    ;WAS FPS CORRECT?
 4841 030566  020405                            CMP     R4,R5
 4842 030570  001402                            BEQ     BB6
 4843 030572  000137  031356                    JMP     BBER0           ;FPS ERROR
 4844                                   ;EXPONENT DIFFERENCE=56=70 (OCT) FD=1
 4845 030576                            BB6:
     030576  012737  030604  001110             MOV     #1$,$LPERR      ;SET UP THE LOOP ON ERROR ADDRESS.
 4846 030604  012704  003200             1$:    MOV     #3200,R4        ;SET FIV,FIV, AND FD
 4847 030610  170104                            LDFPS   R4
 4848 030612  012737  030632  001236             MOV     #BB7,$TMP2
 4849 030620  012700  031740                    MOV     #BBPAT4,R0      ;SET AC0 OPERAND
 4850 030624  172410                            LDD     (R0),AC0
 4851 030626  012700  031710                    MOV     #BBPAT1,R0      ;FSRC
 4852 030632  172010                    BB7:    ADDD    (R0),AC0        ;TEST INSTRUCTION
 4853 030634  170205                            STFPS   R5              ;GET FPS
 4854 030636  012700  031670                    MOV     #BBDAT0,R0      ;GET THE RESULT
 4855 030642  174010                            STD     AC0,(R0)
 4856 030644  012701  032000                    MOV     #BBP10,R1       ;IS IT CORRECT
 4857 030650  012702  000004                    MOV     #4,R2
 4858 030654  022021                    BB10:   CMP     (R0)+,(R1)+
 4859 030656  001415                            BEQ     BB13
 4860 030660  012700  031670                    MOV     #BBDAT0,R0      ;DID A BAD
 4861 030664  012701  031740                    MOV     #BBPAT4,R1      ;CONSTANT (NOT 57)
```

```
 4862 030670  012702  000004                    MOV     #4,R2           ;GET GENERATED
 4863 030674  022021                    BB11:   CMP     (R0)+,(R1)+     ;FOR THE ALIGNMENT
 4864 030676  001402                            BEQ     BB12            ;FLOWS?
 4865 030700  000137  031454                    JMP     BBER2           ;DATA ERROR.D
 4866 030704  077205                    BB12:   SOB     R2,BB11
 4867 030706  000137  031472                    JMP     BBER3           ;BAD CONSTANT.D
 4868 030712  077220                    BB13:   SOB     R2,BB10
 4869 030714  020405                            CMP     R4,R5           ;FPS CORRECT?
 4870 030716  001402                            BEQ     BB14
 4871 030720  000137  031356                    JMP     BBER0           ;BAD FPS.
 4872                                   ;EXPONENT DIFFERENCE=25=31 (OCT) FD=0
 4873 030724                            BB14:
      030724  012737  030732  001110            MOV     #1$,$LPERR      ;SET UP THE LOOP ON ERROR ADDRESS.
 4874 030732  012737  030760  001236  1$:     MOV     #BB15,$TMP2
 4875 030740  012700  031700                    MOV     #BBPAT0,R0      ;SET UP AC0 OPERAND
 4876 030744  172410                            LDD     (R0),AC0
 4877 030746  012704  003000                    MOV     #3000,R4        ;SET FIV AND FIV
 4878                                                                   ;CLEAR FD
 4879 030752  170104                            LDFPS   R4
 4880 030754  012700  031710                    MOV     #BBPAT1,R0      ;FSRC
 4881 030760  172010                    BB15:   ADDF    (R0),AC0        ;TEST INSTRUCTION
 4882 030762  170205                            STFPS   R5
 4883 030764  170011                            SETD                    ;REENTED DOUBLE MODE.
 4884 030766  012700  031670                    MOV     #BBDAT0,R0      ;GET THE RESULT
 4885 030772  174010                            STD     AC0,(R0)
 4886 030774  012701  031700                    MOV     #BBPAT0,R1      ;IS THE RESULT
 4887 031000  012702  000002                    MOV     #2,R2           ;CORRECT?
 4888 031004  022021                    BB16:   CMP     (R0)+,(R1)+
 4889 031006  001402                            BEQ     BB17
 4890 031010  000137  031526                    JMP     BBER4           ;DATA ERROR F
 4891 031014  077205                    BB17:   SOB     R2,BB16
 4892 031016  020405                            CMP     R4,R5           ;IS FPS CORRECT?
 4893 031020  001402                            BEQ     BB20
 4894 031022  000137  031376                    JMP     BBER10          ;FPS ERROR.
 4895                                   ;EXPONENT DIFFERENCE=24=30 (OCT)
 4896 031026                            BB20:
      031026  012737  031034  001110            MOV     #1$,$LPERR      ;SET UP THE LOOP ON ERROR ADDRESS.
 4897 031034  012737  031062  001236  1$:     MOV     #BB21,$TMP2
 4898 031042  012700  031730                    MOV     #BBPAT3,R0      ;SET UP AC0 OPERAND.
 4899 031046  172410                            LDD     (R0),AC0
 4900 031050  012704  003000                    MOV     #3000,R4        ;SET FIU,FIV. CLEAR FD.
 4901 031054  170104                            LDFPS   R4
 4902 031056  012700  031710                    MOV     #BBPAT1,R0      ;FSRC
 4903 031062  172010                    BB21:   ADDF    (R0),AC0        ;TEST INSTRUCTION
 4904 031064  170205                            STFPS   R5
 4905 031066  170011                            SETD                    ;REENTER DOUBLE MODE
 4906 031070  012700  031670                    MOV     #BBDAT0,R0      ;GET THE RESULT
 4907 031074  174010                            STD     AC0,(R0)
 4908 031076  012701  031770                    MOV     #BBP7,R1        ;IS THE RESULT CORRECT?
 4909 031102  012702  000002                    MOV     #2,R2
 4910 031106  022021                    BB22:   CMP     (R0)+,(R1)+
 4911 031110  001415                            BEQ     BB25
 4912 031112  012700  031670                    MOV     #BBDAT0,R0      ;WAS A BAD CONSTANT
 4913 031116  012701  031730                    MOV     #BBPAT3,R1      ;USED (NOT 25) IN
 4914 031122  012702  000002                    MOV     #2,R2           ;THE ALLIGN FLOWS?
 4915 031126  022021                    BB23:   CMP     (R0)+,(R1)+
 4916 031130  001402                            BEQ     BB24
```

```
4917 031132  000137  031562                    JMP     BBER5           ;DATA ERROR F
4918 031136  077205                    BB24:   SOB     R2,BB23
4919 031140  000137  031600                    JMP     BBER6           ;BAD CONSTANT F
4920 031144  077220                    BB25:   SOB     R2,BB22
4921 031146  020405                            CMP     R4,R5
4922 031150  001402                            BEQ     BB26
4923 031152  000137  031376                    JMP     BBER10          ;BAD FPS.
4924                                   ;EXPONENT DIFFERENCE=1
4925 031156                            BB26:
     031156  012737  031164  001110            MOV     #1$,$LPERR      ;SET UP THE LOOP ON ERROR ADDRESS.
4926 031164  012737  031212  001236  1$:     MOV     #BB27,$TMP2
4927 031172  012704  003200                    MOV     #3200,R4
4928 031176  170104                            LDFPS   R4              ;SET UP AC0 OPERAND
4929 031200  012700  031750                    MOV     #BBPAT5,R0
4930 031204  172410                            LDD     (R0),AC0
4931 031206  012700  031710                    MOV     #BBPAT1,R0      ;FSRC
4932 031212  172010                    BB27:   ADDD    (R0),AC0        ;TEST INSTRUCTION
4933 031214  170205                            STFPS   R5
4934 031216  012700  031670                    MOV     #BBDAT0,R0      ;GET THE RESULT.
4935 031222  174010                            STD     AC0,(R0)
4936 031224  012701  032010                    MOV     #BBP11,R1       ;IS IT CORRECT?
4937 031230  012702  000004                    MOV     #4,R2
4938 031234  022021                    BB30:   CMP     (R0)+,(R1)+
4939 031236  001402                            BEQ     BB31
4940 031240  000137  031634                    JMP     BBER7           ;DATA ERROR D
4941 031244  077205                    BB31:   SOB     R2,BB30
4942 031246  020405                            CMP     R4,R5           ;IS FPS CORRECT
4943 031250  001402                            BEQ     BB32
4944 031252  000137  031356                    JMP     BBER0
4945                                   ;EXPONENT DIFFERENCE=100=144 (OCT)
4946 031256                            BB32:
     031256  012737  031264  001110            MOV     #1$,$LPERR      ;SET UP THE LOOP ON ERROR ADDRESS.
4947 031264  012737  031312  001236  1$:     MOV     #BB33,$TMP2
4948 031272  012704  003200                    MOV     #3200,R4
4949 031276  170104                            LDFPS   R4              ;SET FIV,FIV AND FD
4950 031300  012700  031760                    MOV     #BBPAT6,R0      ;SET UP AC0 OPERAND.
4951 031304  172410                            LDD     (R0),AC0
4952 031306  012700  031710                    MOV     #BBPAT1,R0      ;FSRC
4953 031312  172010                    BB33:   ADDD    (R0),AC0        ;TEST INSTRUCTION
4954 031314  170205                            STFPS   R5
4955 031316  012700  031670                    MOV     #BBDAT0,R0      ;GET THE RESULT
4956 031322  174010                            STD     AC0,(R0)
4957 031324  012701  031760                    MOV     #BBPAT6,R1      ;IS IT CORRECT
4958 031330  012702  000004                    MOV     #4,R2
4959 031334  022021                    BB34:   CMP     (R0)+,(R1)+
4960 031336  001402                            BEQ     BB35
4961 031340  000137  031652                    JMP     BBER8           ;DATA ERROR D
4962 031344  077205                    BB35:   SOB     R2,BB34
4963 031346  020405                            CMP     R4,R5           ;IS FPS CORRECT
4964 031350  001002                            BNE     BBER0
4965 031352  000137  032020                    JMP     BBDONE
4966 031356  010437  001242            BBER0:  MOV     R4,$TMP4        ;FPS ERROR D
4967 031362  010537  001240                    MOV     R5,$TMP3
4968 031366  104164                    1$:     ERROR   +164
4969 031370  104413                            RSETUP                  ;GO INITIALIZE THE FPS AND STACK; AND
                                                                       ;SEE IF THE USER HAS EXPRESSED
                                                                       ;THE DESIRE TO CHANGE THE SOFTWARE
```

```
                                                                ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
                                                                ;THE USER TYPED CONTROL G?).
 4970 031372  000137  032020                    JMP     BBDONE
 4971 031376  010437  001242          BBER10:   MOV     R4,$TMP4        ;FPS ERROR F
 4972 031402  010537  001240                    MOV     R5,$TMP3
 4973 031406  104165                 1$:       ERROR   +165
 4974 031410  104413                           RSETUP                  ;GO INITIALIZE THE FPS AND STACK; AND
                                                                ;SEE IF THE USER HAS EXPRESSED
                                                                ;THE DESIRE TO CHANGE THE SOFTWARE
                                                                ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
                                                                ;THE USER TYPED CONTROL G?).
 4975 031412  000137  032020                    JMP     BBDONE
 4976 031416  012737  031720  001242  BBER1:   MOV     #BBPAT2,$TMP4   ;DATA ERROR D
 4977 031424  012737  031720  001246           MOV     #BBPAT2,$TMP6
 4978 031432  012737  031710  001240  BBER11:  MOV     #BBPAT1,$TMP3
 4979 031440  012737  031670  001244           MOV     #BBDAT0,$TMP5
 4980 031446  104166                 1$:       ERROR   +166
 4981 031450  000137  032020                    JMP     BBDONE
 4982 031454  012737  031740  001242  BBER2:   MOV     #BBPAT4,$TMP4
 4983 031462  012737  032000  001246           MOV     #BBP10,$TMP6
 4984 031470  000760                           BR      BBER11
 4985 031472  012737  031740  001242  BBER3:   MOV     #BBPAT4,$TMP4   ;BAD CONSTANT D
 4986 031500  012737  032000  001246           MOV     #BBP10,$TMP6
 4987 031506  012737  031710  001240           MOV     #BBPAT1,$TMP3
 4988 031514  012737  031670  001244           MOV     #BBDAT0,$TMP5
 4989 031522  104167                 1$:       ERROR   +167
 4990 031524  000535                           BR      BBDONE
 4991 031526  012737  031700  001242  BBER4:   MOV     #BBPAT0,$TMP4   ;DATA ERROR F
 4992 031534  012737  031700  001246           MOV     #BBPAT0,$TMP6
 4993 031542  012737  031710  001240  BBER40:  MOV     #BBPAT1,$TMP3
 4994 031550  012737  031670  001244           MOV     #BBDAT0,$TMP5
 4995 031556  104170                 1$:       ERROR   +170
 4996 031560  000517                           BR      BBDONE
 4997 031562  012737  031730  001242  BBER5:   MOV     #BBPAT3,$TMP4
 4998 031570  012737  031770  001246           MOV     #BBP7,$TMP6
 4999 031576  000761                           BR      BBER40
 5000 031600  012737  031730  001242  BBER6:   MOV     #BBPAT3,$TMP4   ;CONSTANT ERROR F
 5001 031606  012737  031770  001246           MOV     #BBP7,$TMP6
 5002 031614  012737  031710  001240           MOV     #BBPAT1,$TMP3
 5003 031622  012737  031670  001244           MOV     #BBDAT0,$TMP5
 5004 031630  104171                 1$:       ERROR   +171
 5005 031632  000472                           BR      BBDONE
 5006 031634  012737  031750  001242  BBER7:   MOV     #BBPAT5,$TMP4
 5007 031642  012737  031710  001246           MOV     #BBPAT11,$TMP6
 5008 031650  000670                           BR      BBER11
 5009 031652  012737  031760  001242  BBER8:   MOV     #BBPAT6,$TMP4
 5010 031660  012737  031760  001246           MOV     #BBPAT6,$TMP6
 5011 031666  000661                           BR      BBER11
 5012 031670  000000  000000  000000  BBDAT0:  .WORD   0,0,0,0
 5013 031700  006400  000000  000000  BBPAT0:  .WORD   6400,0,0,0      ;F(AC)=E(FSRC)+25=26=32(OCT)
 5014 031710  000200  000000  000000  BBPAT1:  .WORD   200,0,0,0       ;E(FSRC)=1
 5015 031720  016400  000000  000000  BBPAT2:  .WORD   16400,0,0,0     ;E(AC)=E(FSRC)+57=58=72(OCT)
 5016 031730  006200  000000  000000  BBPAT3:  .WORD   6200,0,0,0      ;E(AC)=E(FSRC)+24=25=31(OCT)
 5017 031740  016200  000000  000000  BBPAT4:  .WORD   16200,0,0,0     ;E(AC)=E(FSR6)+56=57=71(OCT)
 5018 031750  000400  000000  000000  BBPAT5:  .WORD   400,0,0,0       ;E(AC)=E(FSRC)+1=2
 5019 031760  031200  000000  000000  BBPAT6:  .WORD   31200,0,0,0     ;E(AC)=E(FSRC)+100=101=145(OCT)
 5020 031770  006200  000001  000000  BBP7:    .WORD   6200,1,0,0      ;BBPAT3 RES
```

```
5021 032000  016200  000000  000000  BBP10:  .WORD   16200,0,0,1     ;BBPAT4 RES
5022 032010  000500  000000  000000  BBP11:  .WORD   500,0,0,0       ;BBPAT5 RES
5023 032020                          BBDONE:
     032020  104413                          RSETUP                 ;GO INITIALIZE THE FPS AND STACK; AND
                                                                    ;SEE IF THE USER HAS EXPRESSED
                                                                    ;THE DESIRE TO CHANGE THE SOFTWARE
                                                                    ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
                                                                    ;THE USER TYPED CONTROL G?).
```

```
5031                                    .SBTTL  TEST # 34 - ADDD WITH NEGATVE OPRANDS TEST
                                        ;;*****************************************************************
                                        ;*TEST 34 - ADDD WITH NEGATVE OPRANDS TEST
                                        ;*
                                        ;*THIS IS A TEST OF THE ADDD INSTRUCTION
                                        ;*WITH NEGATIVE OPERANDS.  EVERY COMBINATION OF
                                        ;*OPERAND SIGNS IS TRIED.
                                        ;*
                                        ;;*****************************************************************
     032022  000004                     TST34:  SCOPE
5032                                    ;BOTH OPERANDS NEGATIVE
5033 032024                             DD1:
     032024  012737  032032  001110             MOV     #1$,$LPERR      ;SET UP THE LOOP ON ERROR ADDRESS.
5034 032032  012704  003200             1$:     MOV     #3200,R4                ;SET FIO, FIV, AND FD
5035 032036  170104                             LDFPS   R4
5036 032040  012737  032060  001236             MOV     #DD2,$TMP2
5037 032046  012700  033740                     MOV     #DDP1,R0                ;SET AC0 OPERAND
5038 032052  172410                             LDD     (R0),AC0
5039 032054  012700  033740                     MOV     #DDP1,R0                ;ESRC
5040 032060  172010                     DD2:    ADDD    (R0),AC0                ;TEST INSTRUCTION
5041 032062  170205                             STFPS   R5                      ;GET FPS
5042 032064  012700  033720                     MOV     #DDDAT0,R0              ;GET THE RESULT
5043 032070  174010                             STD     AC0,(R0)
5044 032072  012701  034040                     MOV     #DDP9,R1                ;IS IT CORRECT
5045 032076  012702  000004                     MOV     #4,R2
5046 032102  022021                     DD3:    CMP     (R0)+,(R1)+
5047 032104  001415                             BEQ     DD6
5048 032106  012700  033720                     MOV     #DDDAT0,R0              ;DID A ADD-SUB
5049 032112  012701  033770                     MOV     #DDP4,R1                ;FLOW A FAILURE
5050 032116  012702  000004                     MOV     #4,R2
5051 032122  022021                     DD4:    CMP     (R0)+,(R1)+
5052 032124  001402                             BEQ     DD5                     ;216,442,500
5053 032126  000137  033150                     JMP     DDER1                   ;DATA ERROR,D
5054 032132  077205                     DD5:    SOB     R2,DD4
5055 032134  000137  033206                     JMP     DDER2                   ;FLOW FAILURE,D
5056 032140  077220                     DD6:    SOB     R2,DD3
5057 032142  052704  000010                     BIS     #10,R4
5058 032146  020405                             CMP     R4,R5                   ;FPS CORRECT?
5059 032150  001402                             BEQ     DD7
5060 032152  000137  033132                     JMP     DDER0                   ;BAD,FPS
5061                                    ;AC POS FSRC NEG        AC=-FSRC
5062 032156                             DD7:
     032156  012737  032164  001110             MOV     #1$,$LPERR      ;SET UP THE LOOP ON ERROR ADDRESS.
5063 032164  012704  003200             1$:     MOV     #3200,R4                ;SET FIO, FIV, AND FD
5064 032170  170104                             LDFPS   R4
5065 032172  012737  032212  001236             MOV     #DD8,$TMP2
5066 032200  012700  033750                     MOV     #DDP2,R0                ;SET AC0 OPERAND
5067 032204  172410                             LDD     (R0),AC0
5068 032206  012700  033740                     MOV     #DDP1,R0                ;FSPC
5069 032212  172010                     DD8:    ADDD    (R0),AC0                ;TEST INSTRUCTION
5070 032214  170205                             STFPS   R5                      ;GET FPS
5071 032216  012700  033720                     MOV     #DDDAT0,R0              ;GET THE RESULT
5072 032222  174010                             STD     AC0,(R0)
5073 032224  012701  033730                     MOV     #DDP0,R1                ;IS IT CORRECT
5074 032230  012702  000004                     MOV     #4,R2
5075 032234  022021                     DD10:   CMP     (R0)+,(R1)+
5076 032236  001402                             BEQ     DD11
```

TEST # 34 - ADDD WITH NEGATVE OPRANDS TEST

```
5077 032240  000137  033244                DD11:   JMP     DDER3                   ;FLOW FAILURE
5078 032244  077205                              SOB     R2,DD10
5079 032246  052704  000004                      BIS     #4,R4
5080 032252  020405                              CMP     R4,R5                   ;FPS CORRECT?
5081 032254  001402                              BEQ     DD12
5082 032256  000137  033132                      JMP     DDER0                   ;BAD FPS
5083                                      ;AC NEG FSRC POS                AC=-FSRC
5084 032262                               DD12:
     032262  012737  032270  001110              MOV     #1$,$LPERR      ;SET UP THE LOOP ON ERROR ADDRESS.
5085 032270  012704  003200                1$:   MOV     #3200,R4                ;SET FIU, FIV, AND FD
5086 032274  170104                              LDFPS   R4
5087 032276  012737  032316  001236              MOV     #DD13,$TMP2
5088 032304  012700  033740                      MOV     #DDP1,R0                ;SET AC0 OPERAND
5089 032310  172410                              LDD     (R0),AC0
5090 032312  012700  033750                      MOV     #DDP2,R0                ;FSRC
5091 032316  172010                       DD13:  ADDD    (R0),AC0                ;TEST INSTRUCTION
5092 032320  170205                              STFPS   R5                      ;GET FPS
5093 032322  012700  033720                      MOV     #DDDAT0,R0              ;GET THE RESULT
5094 032326  174010                              STD     AC0,(R0)
5095 032330  012701  033730                      MOV     #DDP0,R1                ;IS IT CORRECT
5096 032334  012702  000004                      MOV     #4,R2
5097 032340  022021                       DD14:  CMP     (R0)+,(R1)+
5098 032342  001402                              BEQ     DD15
5099 032344  000137  033302                      JMP     DDER4                   ;FLOW FAILURE 216,440,121
5100 032350  077205                       DD15:  SOB     R2,DD14
5101 032352  052704  000004                      BIS     #4,R4
5102 032356  020405                              CMP     R4,R5                   ;EPS CORRECT?
5103 032360  001402                              BEQ     DD16
5104 032362  000137  033132                      JMP     DDER0                   ;BAD FPS
5105                                      ;AC0 POC        FSRC    NEG     /AC/ > /FSRC/
5106 032366                               DD16:
     032366  012737  032374  001110              MOV     #1$,$LPERR      ;SET UP THE LOOP ON ERROR ADDRESS.
5107 032374  012704  003200                1$:   MOV     #3200,R4                ;SET FIV, FIV AND FD
5108 032400  170104                              LDFPS   R4
5109 032402  012737  032422  001236              MOV     #DD17,$TMP2
5110 032410  012700  033760                      MOV     #DDP3,R0                ;SET AC0 OPERAND
5111 032414  172410                              LDD     (R0),AC0
5112 032416  012700  034010                      MOV     #DDP6,R0                ;ESPC
5113 032422  172010                       DD17:  ADDD    (R0),AC0                ;TEST INSTRUCTION
5114 032424  170205                              STFPS   R5                      ;GET FPS
5115 032426  012700  033720                      MOV     #DDDAT0,R0              ;GET THE RESULT
5116 032432  174010                              STD     AC0,(R0)
5117 032434  012701  034020                      MOV     #DDP7,R1                ;IS IT CORRECT
5118 032440  012702  000004                      MOV     #4,R2
5119 032444  022021                       DD18:  CMP     (R0)+,(R1)+
5120 032446  001415                              BEQ     DD21
5121 032450  012700  033720                      MOV     #DDDAT0,R0              ;FLOWS FAILURE
5122 032454  012701  034030                      MOV     #DDP8,R1                ;216,440,101
5123 032460  012702  000004                      MOV     #4,R2                   ;GET GENERATED
5124 032464  022021                       DD19:  CMP     (R0)+,(R1)+
5125 032466  001402                              BEQ     DD20
5126 032470  000137  033340                      JMP     DDER5                   ;DATA ERROR.
5127 032474  077205                       DD20:  SOB     R2,DD19
5128 032476  000137  033376                      JMP     DDER6
5129 032502  077220                       DD21:  SOB     R2,DD18
5130 032504  020405                              CMP     R4,R5                   ;EPS CORRECT?
5131 032506  001402                              BEQ     DD22
```

```
5132 032510  000137  033132                    JMP     DDERO                   ;BAD FPS
5133                                ;AC NEG FSRC    POS     /FSRC/ > /AC/
5134 032514                         DD22:
     032514  012737  032522  001110             MOV     #1$,$LPERR      ;SET UP THE LOOP ON ERROR ADDRESS.
5135 032522  012704  003200          1$:        MOV     #3200,R4                ;SET FIO,FIV, AND FD
5136 032526  170104                             LDFPS   R4
5137 032530  012737  032550  001236             MOV     #DD23,$TMP2
5138 032536  012700  034010                     MOV     #DDP6,R0                ;SET ACO OPERAND
5139 032542  172410                             LDD     (R0),AC0
5140 032544  012700  033760                     MOV     #DDP3,R0                ;FSPC
5141 032550  172010                 DD23:       ADDD    (R0),AC0                ;TEST INSTRUCTION
5142 032552  170205                             STFPS   R5                      ;GET FPS
5143 032554  012700  033720                     MOV     #DDDAT0,R0              ;GET THE RESULT
5144 032560  174010                             STD     AC0,(R0)
5145 032562  012701  034020                     MOV     #DDP7,R1                ;IS IT CORRECT?
5146 032566  012702  000004                     MOV     #4,R2
5147 032572  022021                 DD24:       CMP     (R0)+,(R1)+
5148 032574  001415                             BEQ     DD27
5149 032576  012700  033720                     MOV     #DDDAT0,R0              ;FLO,S FAILURE
5150 032602  012701  034030                     MOV     #DDP8,R1                ;CONSTANT (NOT 57)
5151 032606  012702  000004                     MOV     #4,R2                   ;216,042,101
5152 032612  021011                 DD25:       CMP     (R0),(R1)
5153 032614  001402                             BEQ     DD26
5154 032616  000137  033434                     JMP     DDER7                   ;DATA ERROR.
5155 032622  077205                 DD26:       SOB     R2,DD25
5156 032624  000137  033472                     JMP     DDER8
5157 032630  077220                 DD27:       SOB     R2,DD24
5158 032632  020405                             CMP     R4,R5                   ;FPS CORRECT?
5159 032634  001402                             BEQ     DD30
5160 032636  000137  033132                     JMP     DDERO                   ;BAD FPS
5161                                ;AC0 POS        FSRC    NEG     /AC/<//FRSRC/
5162 032642                         DD30:
     032642  012737  032650  001110             MOV     #1$,$LPERR      ;SET UP THE LOOP ON ERROR ADDRESS.
5163 032650  012704  003200          1$:        MOV     #3200,R4                ;SET FIO,FIV,AND FD
5164 032654  170104                             LDFPS   R4
5165 032656  012737  032676  001236             MOV     #DD31,$TMP2
5166 032664  012700  033770                     MOV     #DDP4,R0                ;SET ACO OPERAND
5167 032670  172410                             LDD     (R0),AC0
5168 032672  012700  034000                     MOV     #DDP5,R0                ;FSPC
5169 032676  172010                 DD31:       ADDD    (R0),AC0                ;TEST INSTRUCTION
5170 032700  170205                             STFPS   R5                      ;GET FPS
5171 032702  012700  033720                     MOV     #DDDAT0,R0              ;GET THE RESULT
5172 032706  174010                             STD     AC0,(R0)
5173 032710  012701  034030                     MOV     #DDP8,R1                ;IS IT CORRECT
5174 032714  012702  000004                     MOV     #4,R2
5175 032720  022021                 DD32:       CMP     (R0)+,(R1)+
5176 032722  001415                             BEQ     DD35                    ;ADD-SUB
5177 032724  012700  033720                     MOV     #DDDAT0,R0              ;FLOWAS FAILURE
5178 032730  012701  034020                     MOV     #DDP7,R1                ;CON 216 N440 NOT 141
5179 032734  012702  000004                     MOV     #4,R2                   ;GET GENERATED
5180 032740  022021                 DD33:       CMP     (R0)+,(R1)+             ;FOR THE ALLIGNMENT
5181 032742  001402                             BEQ     DD34                    ;FLOWS?
5182 032744  000137  033530                     JMP     DDER9                   ;DATA ERROR, D
5183 032750  077205                 DD34:       SOB     R2,DD33
5184 032752  000137  033566                     JMP     DDER10
5185 032756  077220                 DD35:       SOB     R2,DD32
5186 032760  052704  000010                     BIS     #10,R4
```

```
5187 032764  020405                        CMP     R4,R5                   ;FPS CORRECT?
5188 032766  001402                        BEQ     DD36
5189 032770  000137  033132                JMP     DDER0                   ;BAD FPS
5190                              ;AC0 NEG        FSRC    POS     /FSRC/</AC/
5191 032774                       DD36:
     032774  012737  033002  001110        MOV     #1$,$LPERR      ;SET UP THE LOOP ON ERROR ADDRESS.
5192 033002  012704  003200          1$:   MOV     #3200,R4                ;SET FIO, FIV, AND FD
5193 033006  170104                        LDFPS   R4
5194 033010  012737  033030  001236        MOV     #DD37,$TMP2
5195 033016  012700  034000                MOV     #DDP5,R0                ;SET AC0 OPERAND
5196 033022  172410                        LDD     (R0),AC0
5197 033024  012700  033770                MOV     #DDP4,R0                ;FSPC
5198 033030  172010                DD37:   ADDD    (R0),AC0                ;TEST INSTRUCTION
5199 033032  170205                        STFPS   R5                      ;GET FPS
5200 033034  012700  033720                MOV     #DDDAT0,R0              ;GET THE RESULT
5201 033040  174010                        STD     AC0,(R0)
5202 033042  012701  034030                MOV     #DDP8,R1                ;IS IT CORRECT
5203 033046  012702  000004                MOV     #4,R2
5204 033052  022021                DD38:   CMP     (R0)+,(R1)+
5205 033054  001415                        BEQ     DD41
5206 033056  012700  033720                MOV     #DDDAT0,R0              ;ADD SUB
5207 033062  012701  034020                MOV     #DDP7,R1                ;FLOWS FAILURES
5208 033066  012702  000004                MOV     #4,R2                   ;GET 216,042,141
5209 033072  022021                DD39:   CMP     (R0)+,(R1)+             ;FOR THE ALLIGNMENT
5210 033074  001402                        BEQ     DD40                    ;FLOWS?
5211 033076  000137  033624                JMP     DDER11                  ;DATA ERROR. D
5212 033102  077205                DD40:   SOB     R2,DD39
5213 033104  000137  033662                JMP     DDER12                  ;BAD CONSTANT.D
5214 033110  077220                DD41:   SOB     R2,DD38
5215 033112  052704  000010                BIS     #10,R4
5216 033116  020405                        CMP     R4,R5                   ;FPS CORRECT?
5217 033120  001402                        BEQ     DD42
5218 033122  000137  033132                JMP     DDER0                   ;BAD FPS
5219 033126  000137  034050          DD42: JMP     DDDONE
5220 033132  010437  001242          DDER0:MOV     R4,$TMP4                ;FPS ERROR
5221 033136  010537  001240                MOV     R5,$TMP3
5222 033142  104164                1$:   ERROR   +164
5223 033144  000137  034050                JMP     DDDONE
5224 033150                        DDER1:
     033150  012737  033740  001240        MOV     #DDP1,$TMP3
     033156  012737  033740  001242        MOV     #DDP1,$TMP4
     033164  012737  033720  001244        MOV     #DDDAT0,$TMP5
     033172  012737  034040  001246        MOV     #DDP9,$TMP6
     033200  104165                1$:   ERROR   +165
     033202  000137  034050                JMP     DDDONE
5225 033206                        DDER2:
     033206  012737  033740  001240        MOV     #DDP1,$TMP3
     033214  012737  033740  001242        MOV     #DDP1,$TMP4
     033222  012737  033720  001244        MOV     #DDDAT0,$TMP5
     033230  012737  034040  001246        MOV     #DDP9,$TMP6
     033236  104176                1$:   ERROR   +176
     033240  000137  034050                JMP     DDDONE
5226 033244                        DDER3:
     033244  012737  033740  001240        MOV     #DDP1,$TMP3
     033252  012737  033750  001242        MOV     #DDP2,$TMP4
     033260  012737  033720  001244        MOV     #DDDAT0,$TMP5
     033266  012737  033730  001246        MOV     #DDP0,$TMP6
```

```
                033274  104177                  1$:     ERROR   +177
                033276  000137  034050                  JMP     DDDONE
          5227  033302                          DDER4:
                033302  012737  033750  001240          MOV     #DDP2,$TMP3
                033310  012737  033740  001242          MOV     #DDP1,$TMP4
                033316  012737  033720  001244          MOV     #DDDAT0,$TMP5
                033324  012737  033730  001246          MOV     #DDP0,$TMP6
                033332  104200                  1$:     ERROR   +200
                033334  000137  034050                  JMP     DDDONE
          5228  033340                          DDER5:
                033340  012737  034010  001240          MOV     #DDP6,$TMP3
                033346  012737  033760  001242          MOV     #DDP3,$TMP4
                033354  012737  033720  001244          MOV     #DDDAT0,$TMP5
                033362  012737  034020  001246          MOV     #DDP7,$TMP6
                033370  104165                  1$:     ERROR   +165
                033372  000137  034050                  JMP     DDDONE
          5229  033376                          DDER6:
                033376  012737  034010  001240          MOV     #DDP6,$TMP3
                033404  012737  033760  001242          MOV     #DDP3,$TMP4
                033412  012737  033720  001244          MOV     #DDDAT0,$TMP5
                033420  012737  034020  001246          MOV     #DDP7,$TMP6
                033426  104201                  1$:     ERROR   +201
                033430  000137  034050                  JMP     DDDONE
          5230  033434                          DDER7:
                033434  012737  033760  001240          MOV     #DDP3,$TMP3
                033442  012737  034010  001242          MOV     #DDP6,$TMP4
                033450  012737  033720  001244          MOV     #DDDAT0,$TMP5
                033456  012737  034020  001246          MOV     #DDP7,$TMP6
                033464  104165                  1$:     ERROR   +165
                033466  000137  034050                  JMP     DDDONE
          5231  033472                          DDER8:
                033472  012737  033760  001240          MOV     #DDP3,$TMP3
                033500  012737  034010  001242          MOV     #DDP6,$TMP4
                033506  012737  033720  001244          MOV     #DDDAT0,$TMP5
                033514  012737  034020  001246          MOV     #DDP7,$TMP6
                033522  104202                  1$:     ERROR   +202
                033524  000137  034050                  JMP     DDDONE
          5232  033530                          DDER9:
                033530  012737  034000  001240          MOV     #DDP5,$TMP3
                033536  012737  033770  001242          MOV     #DDP4,$TMP4
                033544  012737  033720  001244          MOV     #DDDAT0,$TMP5
                033552  012737  034030  001246          MOV     #DDP8,$TMP6
                033560  104165                  1$:     ERROR   +165
                033562  000137  034050                  JMP     DDDONE
          5233  033566                          DDER10:
                033566  012737  034000  001240          MOV     #DDP5,$TMP3
                033574  012737  033770  001242          MOV     #DDP4,$TMP4
                033602  012737  033720  001244          MOV     #DDDAT0,$TMP5
                033610  012737  034030  001246          MOV     #DDP8,$TMP6
                033616  104203                  1$:     ERROR   +203
                033620  000137  034050                  JMP     DDDONE
          5234  033624                          DDER11:
                033624  012737  033770  001240          MOV     #DDP4,$TMP3
                033632  012737  034000  001242          MOV     #DDP5,$TMP4
                033640  012737  033720  001244          MOV     #DDDAT0,$TMP5
                033646  012737  034030  001246          MOV     #DDP8,$TMP6
                033654  104165                  1$:     ERROR   +165
```

```
        033656  000137  034050                  JMP     DDDONE
   5235 033662                          DDER12:
        033662  012737  033770  001240          MOV     #DDP4,$TMP3
        033670  012737  034000  001242          MOV     #DDP5,$TMP4
        033676  012737  033720  001244          MOV     #DDDATO,$TMP5
        033704  012737  034030  001246          MOV     #DDP8,$TMP6
        033712  104204                  1$:     ERROR   +204
        033714  000137  034050                  JMP     DDDONE
   5236 033720  000000  000000  000000  DDDATO: .WORD   0,0,0,0
   5237 033730  000000  000000  000000  DDP0:   .WORD   0,0,0,0
   5238 033740  100200  000000  000000  DDP1:   .WORD   100200,0,0,0    ;-DDP2
   5239 033750  000200  000000  000000  DDP2:   .WORD   200,0,0,0       ;-DDP1
   5240 033760  001100  000000  000000  DDP3:   .WORD   1100,0,0,0      ;EXP=4  ;FRAC=...110...
   5241 033770  000600  000000  000000  DDP4:   .WORD   600,0,0,0       ;EXP=3  ;FRAC=...100...
   5242 034000  101100  000000  000000  DDP5:   .WORD   101100,0,0,0    ;-DDP3
   5243 034010  100600  000000  000000  DDP6:   .WORD   100600,0,0,0    ;-DDP4
   5244 034020  001000  000000  000000  DDP7:   .WORD   1000,0,0,0      ;DDP3+DDP6
   5245 034030  101000  000000  000000  DDP8:   .WORD   101000,0,0,0    ;DDP5+DDP4
   5246 034040  100400  000000  000000  DDP9:   .WORD   100400,0,0,0    ;DDP1+DDP1
   5247 034050                          DDDONE:
        034050  104413                  RSETUP                          ;GO INITIALIZE THE FPS AND STACK; AND
                                                                        ;SEE IF THE USER HAS EXPRESSED
                                                                        ;THE DESIRE TO CHANGE THE SOFTWARE
                                                                        ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
                                                                        ;THE USER TYPED CONTROL G?).
```

```
5255                                    .SBTTL  TEST # 35 - SUBD TEST
                                        ;;**********************************************************************
                                        ;*TEST 35 - SUBD TEST
                                        ;*
                                        ;* THIS IS A TEST OF THE SUBD INSTRUCTION.
                                        ;* BOTH A POSITIVE AND A NEGATIVE NUMBER
                                        ;* IS SUBTRACTED FROM IT SELF
                                        ;*
                                        ;;**********************************************************************
     034052  000004                     TST35:  SCOPE
5256                                    ;       USE POSITIVE OPERANDS
5257 034054                             EE1:
     034054  012737  034062  001110             MOV     #1$,$LPERR      ;SET UP THE LOOP ON ERROR ADDRESS.
5258 034062  012704  003200             1$:     MOV     #3200,R4                ;SET FIO, FIV, AND FD
5259 034066  170104                             LDFPS   R4
5260 034070  012737  034110  001236             MOV     #EE2,$TMP2
5261 034076  012700  034574                     MOV     #EEP1,R0                ;SET AC0 OPERAND
5262 034102  172410                             LDD     (R0),AC0
5263 034104  012700  034574                     MOV     #EEP1,R0                ;FSPC
5264 034110  173010                     EE2:    SUBD    (R0),AC0                ;TEST INSTRUCTION
5265 034112  170205                             STFPS   R5                      ;GET FPS
5266 034114  012700  034552                     MOV     #EEDAT0,R0              ;GET THE RESULT
5267 034120  174010                             STD     AC0,(R0)
5268 034122  012701  034562                     MOV     #EEP0,R1                ;IS IT CORRECT?
5269 034126  012702  000004                     MOV     #4,R2
5270 034132  022021                     EE3:    CMP     (R0)+,(R1)+
5271 034134  001415                             BEQ     EE6
5272 034136  012700  034552                     MOV     #EEDAT0,R0              ;DID A BAD
5273 034142  012701  034604                     MOV     #EEP2,R1                ;CONSTANT (NOT 57)
5274 034146  012702  000004                     MOV     #4,R2                   ;GET GENERATED
5275 034152  022021                     EE4:    CMP     (R0)+,(R1)+             ;FOR THE ALLIGNMENT
5276 034154  001402                             BEQ     EE5                     ;FLOWS?
5277 034156  000137  034362                     JMP     EEER1                   ;DATA ERROR.D
5278 034162  077205                     EE5:    SOB     R2,EE4
5279 034164  000137  034420                     JMP     EEER2                   ;BAD CONSTANT.D
5280 034170  077220                     EE6:    SOB     R2,EE3
5281 034172  052704  000004                     BIS     #4,R4
5282 034176  020405                             CMP     R4,R5                   ;FPS CORRECT?
5283 034200  001402                             BEQ     EE7
5284 034202  000137  034344                     JMP     EEER0                   ;BAD FPS
5285                                    ;USE NEGATIVE OPERANDS
5286 034206                             EE7:
     034206  012737  034214  001110     MOV     #1$,$LPERR      ;SET UP THE LOOP ON ERROR ADDRESS.
5287 034214  012704  003200     1$:     MOV     #3200,R4                ;SET FIO, FIV, AND FD
5288 034220  170104                     LDFPS   R4
5289 034222  012737  034242  001236     MOV     #EE8,$TMP2
5290 034230  012700  034614             MOV     #EEP3,R0                ;SET AC0 OPERAND
5291 034234  172410                     LDD     (R0),AC0
5292 034236  012700  034614             MOV     #EEP3,R0                ;FSPC
5293 034242  173010             EE8:    SUBD    (R0),AC0                ;TEST INSTRUCTION
5294 034244  170205                     STFPS   R5                      ;GET FPS
5295 034246  012700  034552             MOV     #EEDAT0,R0              ;GET THE RESULT
5296 034252  174010                     STD     AC0,(R0)
5297 034254  012701  034562             MOV     #EEP0,R1                ;IS IT CORRECT?
5298 034260  012702  000004             MOV     #4,R2
5299 034264  022021             EE9:    CMP     (R0)+,(R1)+
5300 034266  001415                     BEQ     EE12
```

```
 5301 034270  012700  034552                    MOV     #EEDAT0,R0              ;DID A BAD
 5302 034274  012701  034624                    MOV     #EEP4,R1                ;CONSTANT (NOT 57)
 5303 034300  012702  000004                    MOV     #4,R2                   ;GET GENERATED
 5304 034304  022021                    EE10:   CMP     (R0)+,(R1)+             ;FOR THE ALLIGNMENT
 5305 034306  001402                            BEQ     EE11                    ;FLOWS?
 5306 034310  000137  034456                    JMP     EEER3                   ;DATA ERROR.D
 5307 034314  077205                    EE11:   SOB     R2,EE10
 5308 034316  000137  034514                    JMP     EEER4                   ;BAD CONSTANT.D
 5309 034322  077220                    EE12:   SOB     R2,EE9
 5310 034324  052704  000004                    BIS     #4,R4
 5311 034330  020405                            CMP     R4,R5                   ;FPS CORRECT?
 5312 034332  001402                            BEQ     EE13
 5313 034334  000137  034344                    JMP     EEER0                   ;BAD FPS.
 5314 034340  000137  034634            EE13:   JMP     EEDONE
 5315 034344  010437  001242            EEER0:  MOV     R4,$TMP4                ;BAD FPS
 5316 034350  010537  001240                    MOV     R5,$TMP3
 5317 034354  104205                    1$:     ERROR   +205
 5318 034356  000137  034634                    JMP     EEDONE
 5319 034362                            EEER1:
      034362  012737  034574  001240            MOV     #EEP1,$TMP3
      034370  012737  034574  001242            MOV     #EEP1,$TMP4
      034376  012737  034552  001244            MOV     #EEDAT0,$TMP5
      034404  012737  034562  001246            MOV     #EEP0,$TMP6
      034412  104206                    1$:     ERROR   +206
      034414  000137  034634                    JMP     EEDONE
 5320 034420                            EEER2:
      034420  012737  034574  001240            MOV     #EEP1,$TMP3
      034426  012737  034574  001242            MOV     #EEP1,$TMP4
      034434  012737  034552  001244            MOV     #EEDAT0,$TMP5
      034442  012737  034562  001246            MOV     #EEP0,$TMP6
      034450  104207                    1$:     ERROR   +207
      034452  000137  034634                    JMP     EEDONE
 5321 034456                            EEER3:
      034456  012737  034614  001240            MOV     #EEP3,$TMP3
      034464  012737  034614  001242            MOV     #EEP3,$TMP4
      034472  012737  034552  001244            MOV     #EEDAT0,$TMP5
      034500  012737  034562  001246            MOV     #EEP0,$TMP6
      034506  104206                    1$:     ERROR   +206
      034510  000137  034634                    JMP     EEDONE
 5322 034514                            EEER4:
      034514  012737  034614  001240            MOV     #EEP3,$TMP3
      034522  012737  034614  001242            MOV     #EEP3,$TMP4
      034530  012737  034552  001244            MOV     #EEDAT0,$TMP5
      034536  012737  034562  001246            MOV     #EEP0,$TMP6
      034544  104207                    1$:     ERROR   +207
      034546  000137  034634                    JMP     EEDONE
 5323 034552  000000  000000  000000    EEDAT0: .WORD   0,0,0,0
 5324 034562  000000  000000  000000    EEP0:   .WORD   0,0,0,0,0
 5325 034574  000200  000000  000000    EEP1:   .WORD   200,0,0,0
 5326 034604  000400  000000  000000    EEP2:   .WORD   400,0,0,0
 5327 034614  100200  000000  000000    EEP3:   .WORD   100200,0,0,0
 5328 034624  100400  000000  000000    EEP4:   .WORD   100400,0,0,0
 5329 034634                            EEDONE:
      034634  104413                            RSETUP                  ;GO INITIALIZE THE FPS AND STACK; AND
                                                                        ;SEE IF THE USER HAS EXPRESSED
                                                                        ;THE DESIRE TO CHANGE THE SOFTWARE
                                                                        ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
```

;THE USER TYPED CONTROL G?).

```
5339                                    .SBTTL  TEST # 36 - NORMALIZE ALGORITHM TEST
                                        ;;*****************************************************************
                                        ;*TEST 36 - NORMALIZE ALGORITHM TEST
                                        ;*
                                        ;* THIS IS A TEST OF THE NORMALIZE
                                        ;* FLOW ALGORITHM.  TWO PATTERNS ARE USED,
                                        ;* FIRST THE MINIMUM SITUATION REQUIRING ONE
                                        ;* LEFT SHIFT AND THEN THE MAXIMUM SITUATION
                                        ;* REQUIRING 56 SHIFTS.
                                        ;*
                                        ;;*****************************************************************
     034636  000004                     TST36:  SCOPE
5340                                    ;USE DATA PATTERNS THAT REQUIRE ONLY ONE LEFT SHIFT TO NORMALIZE
5341 034640                             FF1:
     034640  012737  034646  001110             MOV     #1$,$LPERR      ;SET UP THE LOOP ON ERROR ADDRESS.
5342 034646  012704  003200             1$:     MOV     #3200,R4                ;SET FIO, FIV, AND FD
5343 034652  170104                             LDFPS   R4
5344 034654  012737  034674  001236             MOV     #FF2,$TMP2
5345 034662  012700  035172                     MOV     #FFP2,R0                ;SET AC0 OPERAND
5346 034666  172410                             LDD     (R0),AC0
5347 034670  012700  035202                     MOV     #FFP3,R0                ;FSPC
5348 034674  172010                     FF2:    ADDD    (R0),AC0                ;TEST INSTRUCTION
5349 034676  170205                             STFPS   R5                      ;GET FPS
5350 034700  012700  035142                     MOV     #FFDAT0,R0              ;GET THE RESULT
5351 034704  174010                             STD     AC0,(R0)
5352 034706  012701  035212                     MOV     #FFP4, R1               ;IS IT CORRECT
5353 034712  012702  000004                     MOV     #4,R2
5354 034716  022021                     FF3:    CMP     (R0)+,(R1)+
5355 034720  001401                             BEQ     FF4
5356 034722  000470                             BR      FFER2                   ;BAD DATA
5357 034724  077204                     FF4:    SOB     R2,FF3
5358 034726  020405                             CMP     R4,R5                   ;FPS CORRECT?
5359 034730  001401                             BEQ     FF5
5360 034732  000437                             BR      FFER0                   ;BAD FPS
5361                                    ;USE DATA PATTERNS WHICH REQUIRE 56 LEFT SHIFTS TO NORMALIZE
5362                                    ;THE RESULT
5363 034734                             FF5:
     034734  012737  034742  001110             MOV     #1$,$LPERR      ;SET UP THE LOOP ON ERROR ADDRESS.
5364 034742  012704  003200             1$:     MOV     #3200,R4                ;SET FIU, FIV, AND FD
5365 034746  170104                             LDFPS   R4
5366 034750  012737  034770  001236             MOV     #FF6,$TMP2
5367 034756  012700  035152                     MOV     #FFP0,R0                ;SET AC0 OPERAND
5368 034762  172410                             LDD     (R0),AC0
5369 034764  012700  035162                     MOV     #FFP1,R0                ;FSRC
5370 034770  172010                     FF6:    ADDD    (R0),AC0                ;TEST INSTRUCTION
5371 034772  170205                             STFPS   R5                      ;GET FPS
5372 034774  012700  035142                     MOV     #FFDAT0,R0              ;GET THE RESULT
5373 035000  174010                             STD     AC0,(R0)
5374 035002  012701  035212                     MOV     #FFP4,R1                ;IS IT CORRECT
5375 035006  012702  000004                     MOV     #4,R2
5376 035012  022021                     FF7:    CMP     (R0)+,(R1)+
5377 035014  001401                             BEQ     FF10
5378 035016  000413                             BR      FFER1                   ;BATA
5379 035020  077204                     FF10:   SOB     R2,FF7
5380 035022  020405                             CMP     R4,R5                   ;FPS CORRECT?
5381 035024  001401                             BEQ     FF11
5382 035026  000401                             BR      FFER0                   ;BAD FPS
```

```
5383 035030  000474                    FF11:   BR      FFDONE
5384
5385 035032  010537  001240            FFER0:  MOV     R5,$TMP3
5386 035036  010437  001242                    MOV     R4,$TMP4
5387 035042  104164                    1$:     ERROR   +164
5388 035044  000466                            BR      FFDONE
5389
5390 035046                            FFER1:
     035046  012737  035162  001240            MOV     #FFP1,$TMP3
     035054  012737  035152  001242            MOV     #FFP0,$TMP4
     035062  012737  035142  001244            MOV     #FFDAT0,$TMP5
     035070  012737  035212  001246            MOV     #FFP4,$TMP6
     035076  104210                    1$:     ERROR   +210
     035100  000137  035222                    JMP     FFDONE
5391
5392 035104                            FFER2:
     035104  012737  035202  001240            MOV     #FFP3,$TMP3
     035112  012737  035172  001242            MOV     #FFP2,$TMP4
     035120  012737  035142  001244            MOV     #FFDAT0,$TMP5
     035126  012737  035212  001246            MOV     #FFP4,$TMP6
     035134  104210                    1$:     ERROR   +210
     035136  000137  035222                    JMP     FFDONE
5393
5394
5395 035142  000000  000000  000000    FFDAT0: .WORD   0,0,0,0
5396 035152  016000  000000  000000    FFP0:   .WORD   16000,0,0,1
5397 035162  116000  000000  000000    FFP1:   .WORD   116000,0,0,0
5398 035172  000500  000000  000000    FFP2:   .WORD   500,0,0,0
5399 035202  100400  000000  000000    FFP3:   .WORD   100400,0,0,0
5400 035212  000200  000000  000000    FFP4:   .WORD   200,0,0,0       ;FFP4=FFP0+FFP1=FFP3+FFP4
5401 035222                            FFDONE:
5402 035222                            TST37:
```

END OF PASS ROUTINE

```
 5404                                   .SBTTL  END OF PASS ROUTINE
                                        ;;*****************************************************************
                                        ;*INCREMENT THE PASS NUMBER ($PASS)
                                        ;*INDICATE END-OF-PROGRAM AFTER 1 PASSES THRU THE PROGRAM
                                        ;*TYPE "END PASS #XXXXX TOTAL NUMBER OF ERRORS SINCE LAST REPORT YYYYY"
                                        ;*WHERE XXXXX AND YYYYY ARE DECIMAL NUMBERS
                                        ;*IF SW12=1 INHIBIT TRACE TRAP
                                        ;*IF THERES A MONITOR GO TO IT
                                        ;*IF THERE ISN'T JUMP TO LOOP
035222                                  $EOP:
035222  000004                                  SCOPE
035224  005037  001102                         CLR     $TSTNM          ;;ZERO THE TEST NUMBER
035230  005037  001302                         CLR     $TIMES          ;;ZERO THE NUMBER OF ITERATIONS
035234  005237  001324                         INC     $PASS           ;;INCREMENT THE PASS NUMBER
035240  042737  100000  001324                BIC     #100000,$PASS   ;;DON'T ALLOW A NEG. NUMBER
035246  005327                                DEC     (PC)+           ;;LOOP?
035250  000001                          $EOPCT: .WORD   1
035252  003075                                  BGT     $DOAGN          ;;YES
035254  012737                                  MOV     (PC)+,@(PC)+    ;;RESTORE COUNTER
035256  000001                          $ENDCT: .WORD   1
035260  035250                                  $EOPCT
035262  104401  035270                         TYPE    ,65$            ;;TYPE ASCIZ STRING
035266  000407                                  BR      64$             ;;GET OVER THE ASCIZ
                                        ;;65$:  .ASCIZ  <12><15>/END PASS #/
035306                                  64$:
035306  013746  001324                         MOV     $PASS,-(SP)     ;;SAVE $PASS FOR TYPEOUT
                                                                        ;;TYPE PASS NUMBER
035312  104405                                  TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
035314  005737  001112                         TST     $ERTTL          ;&& SEE IF ANY ERRORS THIS PASS
035320  001431                                  BEQ     1000$           ;&& BRANCH AROUND ERROR REPORT IF NONE
035322  104401  035330                         TYPE    ,67$            ;;TYPE ASCIZ STRING
035326  000421                                  BR      66$             ;;GET OVER THE ASCIZ
                                        ;;67$:  .ASCIZ  /  TOTAL ERRORS SINCE LAST REPORT /
035372                                  66$:
035372  013746  001112                         MOV     $ERTTL,-(SP)    ;;SAVE $ERTTL FOR TYPEOUT
                                                                        ;;TOTAL NUMBER OF ERRORS
035376  104405                                  TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
035400  005037  001112                         CLR     $ERTTL          ;;CLEAR ERROR TOTAL
035404  104401  001313                1000$:  TYPE    ,$CRLF          ;;TYPE CARRIAGE RETURN, LINE FEED
035410  013700  000042                $GET42: MOV     @#42,R0         ;;GET MONITOR ADDRESS
035414  001414                                BEQ     $DOAGN          ;;BRANCH IF NO MONITOR
035416  005046                                CLR     -(SP)           ;;INSURE THE "T" BIT IS CLEAR
035420  012746  035426                         MOV     #$CLR.T,-(SP)   ;;SETUP FOR AN RTI OR RTT
035424  000426                                BR      $RTRN           ;;GO DO AN RTI OR RTT TO LOAD THE PSW
                                                                        ;;WITH A CLEARED "T" BIT
035426                                  $CLR.T:
035426  013700  000042                         MOV     @#42,R0         ;;INSURE R0 CONTAINS THE MONITORS
035432  001405                                BEQ     $DOAGN          ;;RETURN ADDRESS
035434  000005                                RESET                   ;;CLEAR THE WORLD
035436  004710                          $ENDAD: JSR     PC,(R0)         ;;GO TO MONITOR
035440  000240                                NOP                     ;;SAVE ROOM
035442  000240                                NOP                     ;;FOR
035444  000240                                NOP                     ;;ACT11
035446                                  $DOAGN:
035446  104400                                  TRAP                    ;;PUSH OLD PSW AND PC ON STACK
035450  042716  000020                         BIC     #20,(SP)        ;;CLEAR THE "T" BIT
035454  032777  010000  143456                BIT     #BIT12,@SWR     ;;RUN WITH TRACE TRAP?
```

END OF PASS ROUTINE

```
035462  001005                          BNE     1$              ;;BR IF NO
035464  005137  035510                  COM     $TBIT           ;;IS IT TIME FOR TRACE TRAP
035470  100402                          BMI     1$              ;;BR IF NO
035472  052716  000020                  BIS     #20,(SP)        ;;SET TRACE TRAP
035476  012746  035504          1$:     MOV     #$LOOP,-(SP)    ;;JUMP TO START OF TEST
035502  000002                  $RTRN:  RTI                     ;;RETURN--THIS IS CHANGED TO
                                                                ;;AN ''RTT'' IF ''RTT'' IS A LEGAL
                                                                ;;INSTRUCTION
035504                          $LOOP:
035504  000137                          JMP     @(PC)+          ;;RETURN
035506  004270                  $RTNAD: .WORD   LOOP
035510  000000                  $TBIT:  .WORD   0               ;;''T'' BIT STATE INDICATOR
035512     377     377     000  $ENULL: .BYTE   -1,-1,0         ;;NULL CHARACTER STRING
                                        .EVEN
```

```
                                .SBTTL  SCOPE HANDLER ROUTINE
                                ;;**********************************************************************
                                ;*THIS ROUTINE CONTROLS THE LOOPING OF SUBTESTS. IT WILL INCREMENT
                                ;*AND LOAD THE TEST NUMBER($TSTNM) INTO THE DISPLAY REG.(DISPLAY<7:0>)
                                ;*AND LOAD THE ERROR FLAG ($ERFLG) INTO DISPLAY<15:08>
                                ;*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 SWR<7:0>
                                ;*CALL
                                ;*      SCOPE           ;;SCOPE=IOT
035516                          $SCOPE:
035516  104407                         CKSWR                   ;;TEST  FOR CHANGE IN SOFT-SWR
035520  032777  040000  143412  1$:     BIT     #BIT14,@SWR     ;;LOOP ON PRESENT TEST?
035526  001114                         BNE     $OVER           ;;YES IF SW14=1
                                ;#####START OF CODE FOR THE XOR TESTER#####
035530  000416                  $XTSTR: BR      6$              ;;IF RUNNING ON THE "XOR" TESTER CHANGE
                                                                ;;THIS INSTRUCTION TO A "NOP" (NOP=240)
035532  013746  000004                  MOV     @#ERRVEC,-(SP)  ;;SAVE THE CONTENTS OF THE ERROR VECTOR
035536  012737  035556  000004         MOV     #5$,@#ERRVEC    ;;SET FOR TIMEOUT
035544  005737  177060                 TST     @#177060        ;;TIME OUT ON XOR?
035550  012637  000004                 MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
035554  000463                         BR      $SVLAD          ;;GO TO THE NEXT TEST
035556  022626                  5$:     CMP     (SP)+,(SP)+     ;;CLEAR THE STACK AFTER A TIME OUT
035560  012637  000004                 MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
035564  000423                         BR      7$              ;;LOOP ON THE PRESENT TEST
035566                          6$:;#####END OF CODE FOR THE XOR TESTER#####
035566  032777  000400  143344         BIT     #BIT08,@SWR     ;;LOOP ON SPEC. TEST?
035574  001404                         BEQ     2$              ;;BR IF NO
035576  127737  143336  001102         CMPB    @SWR,$TSTNM     ;;ON THE RIGHT TEST?    SWR<7:0>
035604  001465                         BEQ     $OVER           ;;BR IF YES
035606  105737  001103          2$:    TSTB    $ERFLG          ;;HAS AN ERROR OCCURRED?
035612  001421                         BEQ     3$              ;;BR IF NO
035614  123737  001115  001103         CMPB    $ERMAX,$ERFLG   ;;MAX. ERRORS FOR THIS TEST OCCURRED?
035622  101015                         BHI     3$              ;;BR IF NO
035624  032777  001000  143306         BIT     #BIT09,@SWR     ;;LOOP ON ERROR?
035632  001404                         BEQ     4$              ;;BR IF NO
035634  013737  001110  001106  7$:    MOV     $LPERR,$LPADR   ;;SET LOOP ADDRESS TO LAST SCOPE
035642  000446                         BR      $OVER
035644  105037  001103          4$:    CLRB    $ERFLG          ;;ZERO THE ERROR FLAG
035650  005037  001302                 CLR     $TIMES          ;;CLEAR THE NUMBER OF ITERATIONS TO MAKE
035654  000415                         BR      1$              ;;ESCAPE TO THE NEXT TEST
035656  032777  004000  143254  3$:    BIT     #BIT11,@SWR     ;;INHIBIT ITERATIONS?
035664  001011                         BNE     1$              ;;BR IF YES
035666  005737  001324                 TST     $PASS           ;;IF FIRST PASS OF PROGRAM
035672  001406                         BEQ     1$              ;;      INHIBIT ITERATIONS
035674  005237  001104                 INC     $ICNT           ;;INCREMENT ITERATION COUNT
035700  023737  001302  001104         CMP     $TIMES,$ICNT    ;;CHECK THE NUMBER OF ITERATIONS MADE
035706  002024                         BGE     $OVER           ;;BR IF MORE ITERATION REQUIRED
035710  012737  000001  001104  1$:    MOV     #1,$ICNT        ;;REINITIALIZE THE ITERATION COUNTER
035716  013737  035774  001302         MOV     $MXCNT,$TIMES   ;;SET NUMBER OF ITERATIONS TO DO
035724  105237  001102          $SVLAD: INCB    $TSTNM          ;;COUNT TEST NUMBERS
035730  113737  001102  001322         MOVB    $TSTNM,$TESTN   ;;SET TEST NUMBER IN APT MAILBOX
035736  011637  001106                 MOV     (SP),$LPADR     ;;SAVE SCOPE LOOP ADDRESS
035742  011637  001110                 MOV     (SP),$LPERR     ;;SAVE ERROR LOOP ADDRESS
035746  005037  001304                 CLR     $ESCAPE         ;;CLEAR THE ESCAPE FROM ERROR ADDRESS
```

```
035752  112737  000001  001115          MOVB    #1,$ERMAX         ;;ONLY ALLOW ONE(1) ERROR ON NEXT TEST
035760  013777  001102  143154  $OVER:  MOV     $TSTNM,@DISPLAY   ;;DISPLAY TEST NUMBER
035766  013716  001106                  MOV     $LPADR,(SP)       ;;FUDGE RETURN ADDRESS
035772  000002                          RTI                       ;;FIXES PS
035774  000001                  $MXCNT: 1                         ;;MAX. NUMBER OF ITERATIONS
```

```
5408                                    .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 ERTYPE ON ERROR
                                        ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
                                        ;*SW15=1         HALT ON ERROR
                                        ;*SW13=1         INHIBIT ERROR TYPEOUTS
                                        ;*SW10=1         BELL ON ERROR
                                        ;*SW09=1         LOOP ON ERROR
                                        ;*CALL
                                        ;*      ERROR   N       ;;ERROR=EMT AND N=ERROR ITEM NUMBER
035776                                  $ERROR:
035776  104407                                  CKSWR                   ;;TEST FOR CHANGE IN SOFT-SWR
036000  105237  001103            7$:     INCB    $ERFLG                  ;;SET THE ERROR FLAG
036004  001775                            BEQ     7$                      ;;DON'T LET THE FLAG GO TO ZERO
036006  013777  001102  143126            MOV     $TSTNM,@DISPLAY ;;DISPLAY TEST NUMBER AND ERROR FLAG
036014  032777  002000  143116            BIT     #BIT10,@SWR             ;;BELL ON ERROR?
036022  001402                            BEQ     1$                      ;;NO - SKIP
036024  104401  001306                    TYPE    ,$BELL                  ;;RING BELL
036030  005237  001112            1$:     INC     $ERTTL                  ;;COUNT THE NUMBER OF ERRORS
036034  011637  001116                    MOV     (SP),$ERRPC             ;;GET ADDRESS OF ERROR INSTRUCTION
036040  162737  000002  001116            SUB     #2,$ERRPC
036046  117737  143044  001114            MOVB    @$ERRPC,$ITEMB  ;;STRIP AND SAVE THE ERROR ITEM CODE
036054  032777  020000  143056            BIT     #BIT13,@SWR             ;;SKIP TYPEOUT IF SET
036062  001004                            BNE     20$                     ;;SKIP TYPEOUTS
036064  004737  040510                    JSR     PC,ERTYPE               ;;GO TO USER ERROR ROUTINE
036070  104401  001313                    TYPE    ,$CRLF
036074                                  20$:
036074  122737  000001  001336                    CMPB    #APTENV,$ENV    ;;RUNNING IN APT MODE
036102  001007                                    BNE     2$              ;;NO,SKIP APT ERROR REPORT
036104  113737  001114  036116                    MOVB    $ITEMB,21$      ;;SET ITEM NUMBER AS ERROR NUMBER
036112  004737  037352                            JSR     PC,$ATY4        ;;REPORT FATAL ERROR TO APT
036116     000                          21$:     .BYTE   0
036117     000                                   .BYTE   0
036120  000777                          22$:     BR      22$             ;;APT ERROR LOOP
036122  005777  143012                  2$:      TST     @SWR            ;;HALT ON ERROR
036126  100002                                   BPL     3$              ;;SKIP IF CONTINUE
036130  000000                                   HALT                    ;;HALT ON ERROR!
036132  104407                                   CKSWR                   ;;TEST FOR CHANGE IN SOFT-SWR
036134  032777  001000  142776            3$:     BIT     #BIT09,@SWR     ;;LOOP ON ERROR SWITCH SET?
036142  001402                                   BEQ     4$              ;;BR IF NO
036144  013716  001110                           MOV     $LPERR,(SP)     ;;FUDGE RETURN FOR LOOPING
036150  005737  001304                    4$:     TST     $ESCAPE         ;;CHECK FOR AN ESCAPE ADDRESS
036154  001402                                   BEQ     5$              ;;BR IF NONE
036156  013716  001304                           MOV     $ESCAPE,(SP)    ;;FUDGE RETURN ADDRESS FOR ESCAPE
036162                                  5$:
036162  022737  035436  000042                   CMP     #$ENDAD,@#42    ;;ACT-11 AUTO-ACCEPT?
036170  001001                                   BNE     6$              ;;BRANCH IF NO
036172  000000                                   HALT                    ;;YES
036174                                  6$:
036174  032777  001000  142736                   BIT     #BIT09,@SWR
036202  001013                                   BNE     ERM10
036204  011637  001162                           MOV     (SP),$REG0              ;SEE IF ERROR #377
036210  062737  177776  001162                   ADD     #-2,$REG0
036216  122777  000377  142736                   CMPB    #377,@$REG0
036224  001002                                   BNE     ERM10
```

```
036226  062716  000002                  ADD     #2,(SP)
036232  000002                  ERM10:  RTI
```

```
 5410                                   .SBTTL  SAVE AND RESTORE R0-R5 ROUTINES
                                        ;;*****************************************************************
                                        ;*SAVE R0-R5
                                        ;*CALL:
                                        ;*      SAVREG
                                        ;*UPON RETURN FROM $SAVREG THE STACK WILL LOOK LIKE:
                                        ;*
                                        ;*TOP---(+16)
                                        ;* +2---(+18)
                                        ;* +4---R5
                                        ;* +6---R4
                                        ;* +8---R3
                                        ;*+10---R2
                                        ;*+12---R1
                                        ;*+14---R0
 036234                                 $SAVREG:
 036234  010046                                 MOV     R0,-(SP)        ;;PUSH R0 ON STACK
 036236  010146                                 MOV     R1,-(SP)        ;;PUSH R1 ON STACK
 036240  010246                                 MOV     R2,-(SP)        ;;PUSH R2 ON STACK
 036242  010346                                 MOV     R3,-(SP)        ;;PUSH R3 ON STACK
 036244  010446                                 MOV     R4,-(SP)        ;;PUSH R4 ON STACK
 036246  010546                                 MOV     R5,-(SP)        ;;PUSH R5 ON STACK
 036250  016646  000022                         MOV     22(SP),-(SP)    ;;SAVE PS OF MAIN FLOW
 036254  016646  000022                         MOV     22(SP),-(SP)    ;;SAVE PC OF MAIN FLOW
 036260  016646  000022                         MOV     22(SP),-(SP)    ;;SAVE PS OF CALL
 036264  016646  000022                         MOV     22(SP),-(SP)    ;;SAVE PC OF CALL
 036270  000002                               RTI
                                        ;*RESTORE R0-R5
                                        ;*CALL:
                                        ;*      RESREG
 036272                                 $RESREG:
 036272  012666  000022                         MOV     (SP)+,22(SP)    ;;RESTORE PC OF CALL
 036276  012666  000022                         MOV     (SP)+,22(SP)    ;;RESTORE PS OF CALL
 036302  012666  000022                         MOV     (SP)+,22(SP)    ;;RESTORE PC OF MAIN FLOW
 036306  012666  000022                         MOV     (SP)+,22(SP)    ;;RESTORE PS OF MAIN FLOW
 036312  012605                               MOV     (SP)+,R5        ;;POP STACK INTO R5
 036314  012604                               MOV     (SP)+,R4        ;;POP STACK INTO R4
 036316  012603                               MOV     (SP)+,R3        ;;POP STACK INTO R3
 036320  012602                               MOV     (SP)+,R2        ;;POP STACK INTO R2
 036322  012601                               MOV     (SP)+,R1        ;;POP STACK INTO R1
 036324  012600                               MOV     (SP)+,R0        ;;POP STACK INTO R0
 036326  000002                               RTI
```

```
5412                                    .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
                                        ;*
036330  105737  001157          $TYPE:  TSTB    $TPFLG           ;;IS THERE A TERMINAL?
036334  100002                          BPL     1$               ;;BR IF YES
036336  000000                          HALT                     ;;HALT HERE IF NO TERMINAL
036340  000430                          BR      3$               ;;LEAVE
036342  010046                  1$:     MOV     R0,-(SP)         ;;SAVE R0
036344  017600  000002                  MOV     @2(SP),R0        ;;GET ADDRESS OF ASCIZ STRING
036350  122737  000001  001336          CMPB    #APTENV,$ENV     ;;RUNNING IN APT MODE
036356  001011                          BNE     62$              ;;NO,GO CHECK FOR APT CONSOLE
036360  132737  000100  001337          BITB    #APTSPOOL,$ENVM  ;;SPOOL MESSAGE TO APT
036366  001405                          BEQ     62$              ;;NO,GO CHECK FOR CONSOLE
036370  010037  036400                  MOV     R0,61$           ;;SETUP MESSAGE ADDRESS FOR APT
036374  004737  037342                  JSR     PC,$ATY3         ;;SPOOL MESSAGE TO APT
036400  000000                  61$:    .WORD   0                ;;MESSAGE ADDRESS
036402  132737  000040  001337  62$:    BITB    #APTCSUP,$ENVM   ;;APT CONSOLE SUPPRESSED
036410  001003                          BNE     60$              ;;YES,SKIP TYPE OUT
036412  112046                  2$:     MOVB    (R0)+,-(SP)      ;;PUSH CHARACTER TO BE TYPED ONTO STACK
036414  001005                          BNE     4$               ;;BR IF IT ISN'T THE TERMINATOR
036416  005726                          TST     (SP)+            ;;IF TERMINATOR POP IT OFF THE STACK
036420  012600                  60$:    MOV     (SP)+,R0         ;;RESTORE R0
036422  062716  000002          3$:     ADD     #2,(SP)          ;;ADJUST RETURN PC
036426  000002                          RTI                      ;;RETURN
036430  122716  000011          4$:     CMPB    #HT,(SP)         ;;BRANCH IF <HT>
036434  001430                          BEQ     8$
036436  122716  000200                  CMPB    #CRLF,(SP)       ;;BRANCH IF NOT <CRLF>
036442  001006                          BNE     5$
036444  005726                          TST     (SP)+            ;;POP  <CR><LF> EQUIV
036446  104401                          TYPE                     ;;TYPE A CR AND LF
036450  001313                          $CRLF
036452  105037  036656                  CLRB    $CHARCNT         ;;CLEAR CHARACTER COUNT
036456  000755                          BR      2$               ;;GET NEXT CHARACTER
036460  004737  036542          5$:     JSR     PC,$TYPEC        ;;GO TYPE THIS CHARACTER
036464  123726  001156          6$:     CMPB    $FILLC,(SP)+     ;;IS IT TIME FOR FILLER CHARS.?
036470  001350                          BNE     2$               ;;IF NO GO GET NEXT CHAR.
036472  013746  001154                  MOV     $NULL,-(SP)      ;;GET # OF FILLER CHARS. NEEDED
                                                                 ;;AND THE NULL CHAR.
036476  105366  000001          7$:     DECB    1(SP)            ;;DOES A NULL NEED TO BE TYPED?
036502  002770                          BLT     6$               ;;BR IF NO--GO POP THE NULL OFF OF STACK
036504  004737  036542                  JSR     PC,$TYPEC        ;;GO TYPE A NULL
036510  105337  036656                  DECB    $CHARCNT         ;;DO NOT COUNT AS A COUNT
036514  000770                          BR      7$               ;;LOOP
                                        ;HORIZONTAL TAB PROCESSOR
036516  112716  000040          8$:     MOVB    #' ,(SP)         ;;REPLACE TAB WITH SPACE
```

TYPE ROUTINE

```
036522  004737  036542          9$:     JSR     PC,$TYPEC       ;;TYPE A SPACE
036526  132737  000007  036656          BITB    #7,$CHARCNT     ;;BRANCH IF NOT AT
036534  001372                          BNE     9$              ;;TAB STOP
036536  005726                          TST     (SP)+           ;;POP SPACE OFF STACK
036540  000724                          BR      2$              ;;GET NEXT CHARACTER
036542  105777  142402          $TYPEC: TSTB    @$TPS           ;;WAIT UNTIL PRINTER IS READY
036546  100375                          BPL     $TYPEC
036550  116677  000002  142374          MOVB    2(SP),@$TPB     ;;LOAD CHAR TO BE TYPED INTO DATA REG.
036556  105777  142362                  TSTB    @$TKS           ;;SEE IF KEYBOARD IS TALKING.
036562  100021                          BPL     2$              ;;BRANCH IF IT ISN'T.
036564  017746  142356                  MOV     @$TKB,-(SP)     ;;PUSH CHARACTER ONTO STACK.
036570  042716  177600                  BIC     #177600,(SP)    ;;BIT CLEAR TOP BYTE AND PARITY BIT.
036574  022726  000023                  CMP     #23,(SP)+       ;;SEE IF THIS IS A ^S.
036600  001012                          BNE     2$              ;;BRANCH TO CONTINUE IF IT ISN'T.
036602  105777  142336          3$:     TSTB    @$TKS           ;;WAIT FOR ANOTHER INPUT.
036606  100375                          BPL     3$              ;;BRANCH BACK IF NOT READY.
036610  017746  142332                  MOV     @$TKB,-(SP)     ;;PUSH NEXT CHARACTER ON STACK.
036614  042716  177600                  BIC     #177600,(SP)    ;;BIT CLEAR TOP BYTE AND PARITY BIT.
036620  022726  000021                  CMP     #21,(SP)+       ;;SEE IF THIS IS A ^Q.
036624  001366                          BNE     3$              ;;BRANCH BACK FOR MORE WAIT IF NOT.
036626  122766  000015  000002  2$:     CMPB    #CR,2(SP)       ;;IS CHARACTER A CARRIAGE RETURN?
036634  001003                          BNE     1$              ;;BRANCH IF NO
036636  105037  036656                  CLRB    $CHARCNT        ;;YES--CLEAR CHARACTER COUNT
036642  000406                          BR      $TYPEX          ;;EXIT
036644  122766  000012  000002  1$:     CMPB    #LF,2(SP)       ;;IS CHARACTER A LINE FEED?
036652  001402                          BEQ     $TYPEX          ;;BRANCH IF YES
036654  105227                          INCB    (PC)+           ;;COUNT THE CHARACTER
036656  000000          $CHARCNT:.WORD  0               ;;CHARACTER COUNT STORAGE
036660  000207          $TYPEX: RTS     PC
```

```
5414                                    .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
036662  017646  000000        $TYPOS: MOV     @(SP),-(SP)     ;;PICKUP THE MODE
036666  116637  000001 037105          MOVB    1(SP),$OFILL    ;;LOAD ZERO FILL SWITCH
036674  112637  037107                MOVB    (SP)+,$OMODE+1  ;;NUMBER OF DIGITS TO TYPE
036700  062716  000002                ADD     #2,(SP)         ;;ADJUST RETURN ADDRESS
036704  000406                        BR      $TYPON
036706  112737  000001 037105  $TYPOC: MOVB    #1,$OFILL       ;;SET THE ZERO FILL SWITCH
036714  112737  000006 037107          MOVB    #6,$OMODE+1     ;;SET FOR SIX(6) DIGITS
036722  112737  000005 037104  $TYPON: MOVB    #5,$OCNT        ;;SET THE ITERATION COUNT
036730  010346                        MOV     R3,-(SP)        ;;SAVE R3
036732  010446                        MOV     R4,-(SP)        ;;SAVE R4
036734  010546                        MOV     R5,-(SP)        ;;SAVE R5
036736  113704  037107                MOVB    $OMODE+1,R4     ;;GET THE NUMBER OF DIGITS TO TYPE
036742  005404                        NEG     R4
036744  062704  000006                ADD     #6,R4           ;;SUBTRACT IT FOR MAX. ALLOWED
036750  110437  037106                MOVB    R4,$OMODE       ;;SAVE IT FOR USE
036754  113704  037105                MOVB    $OFILL,R4       ;;GET THE ZERO FILL SWITCH
036760  016605  000012                MOV     12(SP),R5       ;;PICKUP THE INPUT NUMBER
036764  005003                        CLR     R3              ;;CLEAR THE OUTPUT WORD
036766  006105              1$:       ROL     R5              ;;ROTATE MSB INTO "C"
036770  000404                        BR      3$              ;;GO DO MSB
036772  006105              2$:       ROL     R5              ;;FORM THIS DIGIT
036774  006105                        ROL     R5
036776  006105                        ROL     R5
037000  010503                        MOV     R5,R3
037002  006103              3$:       ROL     R3              ;;GET LSB OF THIS DIGIT
037004  105337  037106                DECB    $OMODE          ;;TYPE THIS DIGIT?
037010  100016                        BPL     7$              ;;BR IF NO
037012  042703  177770                BIC     #177770,R3      ;;GET RID OF JUNK
037016  001002                        BNE     4$              ;;TEST FOR 0
037020  005704                        TST     R4              ;;SUPPRESS THIS 0?
037022  001403                        BEQ     5$              ;;BR IF YES
037024  005204              4$:       INC     R4              ;;DON'T SUPPRESS ANYMORE 0'S
037026  052703  000060                BIS     #'0,R3          ;;MAKE THIS DIGIT ASCII
037032  052703  000040              5$:       BIS     #' ,R3          ;;MAKE ASCII IF NOT ALREADY
```

```
037036  110337  037102                  MOVB    R3,8$           ;;SAVE FOR TYPING
037042  104401  037102                  TYPE    ,8$             ;;GO TYPE THIS DIGIT
037046  105337  037104          7$:     DECB    $OCNT           ;;COUNT BY 1
037052  003347                          BGT     2$              ;;BR IF MORE TO DO
037054  002402                          BLT     6$              ;;BR IF DONE
037056  005204                          INC     R4              ;;INSURE LAST DIGIT ISN'T A BLANK
037060  000744                          BR      2$              ;;GO DO THE LAST DIGIT
037062  012605                  6$:     MOV     (SP)+,R5        ;;RESTORE R5
037064  012604                          MOV     (SP)+,R4        ;;RESTORE R4
037066  012603                          MOV     (SP)+,R3        ;;RESTORE R3
037070  016666  000002  000004          MOV     2(SP),4(SP)     ;;SET THE STACK FOR RETURNING
037076  012616                          MOV     (SP)+,(SP)
037100  000002                          RTI                     ;;RETURN
037102     000                  8$:     .BYTE   0               ;;STORAGE FOR ASCII DIGIT
037103     000                          .BYTE   0               ;;TERMINATOR FOR TYPE ROUTINE
037104     000                  $OCNT:  .BYTE   0               ;;OCTAL DIGIT COUNTER
037105     000                  $OFILL: .BYTE   0               ;;ZERO FILL SWITCH
037106  000000                  $OMODE: .WORD   0               ;;NUMBER OF DIGITS TO TYPE
```

```
 5416                                   .SBTTL  CONVERT BINARY TO DECIMAL AND TYPE ROUTINE
                                        ;;**********************************************************************
                                        ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 5-DIGIT
                                        ;*SIGNED DECIMAL (ASCII) NUMBER AND TYPE IT. DEPENDING ON WHETHER THE
                                        ;*NUMBER IS POSITIVE OR NEGATIVE A SPACE OR A MINUS SIGN WILL BE TYPED
                                        ;*BEFORE THE FIRST DIGIT OF THE NUMBER. LEADING ZEROS WILL ALWAYS BE
                                        ;*REPLACED WITH SPACES.
                                        ;*CALL:
                                        ;*      MOV     NUM,-(SP)       ;;PUT THE BINARY NUMBER ON THE STACK
                                        ;*      TYPDS                   ;;GO TO THE ROUTINE
 037110                                 $TYPDS:
 037110  010046                         MOV     R0,-(SP)        ;;PUSH R0 ON STACK
 037112  010146                         MOV     R1,-(SP)        ;;PUSH R1 ON STACK
 037114  010246                         MOV     R2,-(SP)        ;;PUSH R2 ON STACK
 037116  010346                         MOV     R3,-(SP)        ;;PUSH R3 ON STACK
 037120  010546                         MOV     R5,-(SP)        ;;PUSH R5 ON STACK
 037122  012746  020200                 MOV     #20200,-(SP)    ;;SET BLANK SWITCH AND SIGN
 037126  016605  000020                 MOV     20(SP),R5       ;;GET THE INPUT NUMBER
 037132  100004                         BPL     1$              ;;BR IF INPUT IS POS.
 037134  005405                         NEG     R5              ;;MAKE THE BINARY NUMBER POS.
 037136  112766  000055  000001         MOVB    #'-,1(SP)       ;;MAKE THE ASCII NUMBER NEG.
 037144  005000                 1$:     CLR     R0              ;;ZERO THE CONSTANTS INDEX
 037146  012703  037324                 MOV     #$DBLK,R3       ;;SETUP THE OUTPUT POINTER
 037152  112723  000040                 MOVB    #' ,(R3)+       ;;SET THE FIRST CHARACTER TO A BLANK
 037156  005002                 2$:     CLR     R2              ;;CLEAR THE BCD NUMBER
 037160  016001  037314                 MOV     $DTBL(R0),R1    ;;GET THE CONSTANT
 037164  160105                 3$:     SUB     R1,R5           ;;FORM THIS BCD DIGIT
 037166  002402                         BLT     4$              ;;BR IF DONE
 037170  005202                         INC     R2              ;;INCREASE THE BCD DIGIT BY 1
 037172  000774                         BR      3$
 037174  060105                 4$:     ADD     R1,R5           ;;ADD BACK THE CONSTANT
 037176  005702                         TST     R2              ;;CHECK IF BCD DIGIT=0
 037200  001002                         BNE     5$              ;;FALL THROUGH IF 0
 037202  105716                         TSTB    (SP)            ;;STILL DOING LEADING 0'S?
 037204  100407                         BMI     7$              ;;BR IF YES
 037206  106316                 5$:     ASLB    (SP)            ;;MSD?
 037210  103003                         BCC     6$              ;;BR IF NO
 037212  116663  000001  177777         MOVB    1(SP),-1(R3)    ;;YES--SET THE SIGN
 037220  052702  000060         6$:     BIS     #'0,R2          ;;MAKE THE BCD DIGIT ASCII
 037224  052702  000040         7$:     BIS     #' ,R2          ;;MAKE IT A SPACE IF NOT ALREADY A DIGIT
 037230  110223                         MOVB    R2,(R3)+        ;;PUT THIS CHARACTER IN THE OUTPUT BUFFER
 037232  005720                         TST     (R0)+           ;;JUST INCREMENTING
 037234  020027  000010                 CMP     R0,#10          ;;CHECK THE TABLE INDEX
 037240  002746                         BLT     2$              ;;GO DO THE NEXT DIGIT
 037242  003002                         BGT     8$              ;;GO TO EXIT
 037244  010502                         MOV     R5,R2           ;;GET THE LSD
 037246  000764                         BR      6$              ;;GO CHANGE TO ASCII
 037250  105726                 8$:     TSTB    (SP)+           ;;WAS THE LSD THE FIRST NON-ZERO?
 037252  100003                         BPL     9$              ;;BR IF NO
 037254  116663  177777  177776         MOVB    -1(SP),-2(R3)   ;;YES--SET THE SIGN FOR TYPING
 037262  105013                 9$:     CLRB    (R3)            ;;SET THE TERMINATOR
 037264  012605                         MOV     (SP)+,R5        ;;POP STACK INTO R5
 037266  012603                         MOV     (SP)+,R3        ;;POP STACK INTO R3
 037270  012602                         MOV     (SP)+,R2        ;;POP STACK INTO R2
 037272  012601                         MOV     (SP)+,R1        ;;POP STACK INTO R1
 037274  012600                         MOV     (SP)+,R0        ;;POP STACK INTO R0
 037276  104401  037324                 TYPE    ,$DBLK          ;;NOW TYPE THE NUMBER
```

CONVERT BINARY TO DECIMAL AND TYPE ROUTINE

```
037302  016666  000002  000004          MOV     2(SP),4(SP)     ;;ADJUST THE STACK
037310  012616                          MOV     (SP)+,(SP)
037312  000002                          RTI                     ;;RETURN TO USER
037314  023420                  $DTBL:  10000.
037316  001750                          1000.
037320  000144                          100.
037322  000012                          10.
037324                          $DBLK:  .BLKW   4
```

```
5418                                    .SBTTL  APT COMMUNICATIONS ROUTINE
                                        ;;**********************************************************************
037334  112737  000001  037600  $ATY1:  MOVB    #1,$FFLG        ;;TO REPORT FATAL ERROR
037342  112737  000001  037576  $ATY3:  MOVB    #1,$MFLG        ;;TO TYPE A MESSAGE
037350  000403                  $ATY2:  BR      $ATYC
037352  112737  000001  037600  $ATY4:  MOVB    #1,$FFLG        ;;TO ONLY REPORT FATAL ERROR
037360                          $ATYC:
037360  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
037362  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
037364  105737  037576                  TSTB    $MFLG           ;;SHOULD TYPE A MESSAGE?
037370  001450                          BEQ     5$              ;;IF NOT:  BR
037372  122737  000001  001336          CMPB    #APTENV,$ENV    ;;OPERATING UNDER APT?
037400  001031                          BNE     3$              ;;IF NOT:  BR
037402  132737  000100  001337          BITB    #APTSPOOL,$ENVM ;;SHOULD SPOOL MESSAGES?
037410  001425                          BEQ     3$              ;;IF NOT:  BR
037412  017600  000004                  MOV     @4(SP),R0       ;;GET MESSAGE ADDR.
037416  062766  000002  000004          ADD     #2,4(SP)                ;;BUMP RETURN ADDR.
037424  005737  001316          1$:     TST     $MSGTYPE        ;;SEE IF DONE W/ LAST XMISSION?
037430  001375                          BNE     1$              ;;IF NOT:  WAIT
037432  010037  001332                  MOV     R0,$MSGAD       ;;PUT ADDR IN MAILBOX
037436  105720                  2$:     TSTB    (R0)+           ;;FIND END OF MESSAGE
037440  001376                          BNE     2$
037442  163700  001332                  SUB     $MSGAD,R0       ;;SUB START OF MESSAGE
037446  006200                          ASR     R0              ;;GET MESSAGE LNGTH IN WORDS
037450  010037  001334                  MOV     R0,$MSGLGT      ;;PUT LENGTH IN MAILBOX
037454  012737  000004  001316          MOV     #4,$MSGTYPE     ;;TELL APT TO TAKE MSG.
037462  000413                          BR      5$
037464  017637  000004  037510  3$:     MOV     @4(SP),4$       ;;PUT MSG ADDR IN JSR LINKAGE
037472  062766  000002  000004          ADD     #2,4(SP)                ;;BUMP RETURN ADDRESS
037500  013746  177776                  MOV     177776,-(SP)    ;;PUSH 177776 ON STACK
037504  004737  036330                  JSR     PC,$TYPE        ;;CALL TYPE MACRO
037510  000000                  4$:     .WORD   0
037512                          5$:
037512  105737  037600          10$:    TSTB    $FFLG           ;;SHOULD REPORT FATAL ERROR?
037516  001416                          BEQ     12$             ;;IF NOT:  BR
037520  005737  001336                  TST     $ENV            ;;RUNNING UNDER APT?
037524  001413                          BEQ     12$             ;;IF NOT:  BR
037526  005737  001316          11$:    TST     $MSGTYPE        ;;FINISHED LAST MESSAGE?
037532  001375                          BNE     11$             ;;IF NOT:  WAIT
037534  017637  000004  001320          MOV     @4(SP),$FATAL   ;;GET ERROR #
037542  062766  000002  000004          ADD     #2,4(SP)                ;;BUMP RETURN ADDR.
037550  005237  001316                  INC     $MSGTYPE        ;;TELL APT TO TAKE ERROR
037554  105037  037600          12$:    CLRB    $FFLG           ;;CLEAR FATAL FLAG
037560  105037  037577                  CLRB    $LFLG           ;;CLEAR LOG FLAG
037564  105037  037576                  CLRB    $MFLG           ;;CLEAR MESSAGE FLAG
037570  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
037572  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
037574  000207                          RTS     PC              ;;RETURN
037576     000          $MFLG:  .BYTE   0               ;;MESSG. FLAG
037577     000          $LFLG:  .BYTE   0               ;;LOG FLAG
037600     000          $FFLG:  .BYTE   0               ;;FATAL FLAG
                                .EVEN
            000200      APTSIZE=200
            000001      APTENV=001
            000100      APTSPOOL=100
            000040      APTCSUP=040
```

```
5420                                    .SBTTL  TTY INPUT ROUTINE
                                        ;;**********************************************************************
                                        .ENABL  LSB
                                        ;;**********************************************************************
                                        ;*SOFTWARE SWITCH REGISTER CHANGE ROUTINE.
                                        ;*ROUTINE IS ENTERED FROM THE TRAP HANDLER, AND WILL
                                        ;*SERVICE THE TEST FOR CHANGE IN SOFTWARE SWITCH REGISTER TRAP CALL
                                        ;*WHEN OPERATING IN TTY FLAG MODE.
037602  022737  000176  001140  $CKSWR: CMP     #SWREG,SWR      ;;IS THE SOFT-SWR SELECTED?
037610  001074                         BNE     15$             ;;BRANCH IF NO
037612  105777  141326                 TSTB    @$TKS           ;;CHAR THERE?
037616  100071                         BPL     15$             ;;IF NO, DON'T WAIT AROUND
037620  117746  141322                 MOVB    @$TKB,-(SP)     ;;SAVE THE CHAR
037624  042716  177600                 BIC     #^C177,(SP)     ;;STRIP-OFF THE ASCII
037630  022726  000007                 CMP     #7,(SP)+        ;;IS IT A CONTROL G?
037634  001062                         BNE     15$             ;;NO, RETURN TO USER
037636  123727  001134  000001         CMPB    $AUTOB,#1       ;;ARE WE RUNNING IN AUTO-MODE?
037644  001456                         BEQ     15$             ;;BRANCH IF YES
037646  104401  040211                 TYPE    ,$CNTLG         ;;ECHO THE CONTROL-G (^G)
037652  104401  040216         $GTSWR: TYPE    ,$MSWR          ;;TYPE CURRENT CONTENTS
037656  013746  000176                 MOV     SWREG,-(SP)     ;;SAVE SWREG FOR TYPEOUT
037662  104402                         TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
037664  104401  040227                 TYPE    ,$MNEW          ;;PROMPT FOR NEW SWR
037670  005046                 19$:    CLR     -(SP)           ;;CLEAR COUNTER
037672  005046                         CLR     -(SP)           ;;THE NEW SWR
037674  105777  141244          7$:    TSTB    @$TKS           ;;CHAR THERE?
037700  100375                         BPL     7$              ;;IF NOT TRY AGAIN
037702  117746  141240                 MOVB    @$TKB,-(SP)     ;;PICK UP CHAR
037706  042716  177600                 BIC     #^C177,(SP)     ;;MAKE IT 7-BIT ASCII
037712  021627  000025          9$:    CMP     (SP),#25        ;;IS IT A CONTROL-U?
037716  001005                         BNE     10$             ;;BRANCH IF NOT
037720  104401  040204                 TYPE    ,$CNTLU         ;;YES, ECHO CONTROL-U (^U)
037724  062706  000006          20$:   ADD     #6,SP           ;;IGNORE PREVIOUS INPUT
037730  000757                         BR      19$             ;;LET'S TRY IT AGAIN
037732  021627  000015          10$:   CMP     (SP),#15        ;;IS IT A <CR>?
037736  001022                         BNE     16$             ;;BRANCH IF NO
037740  005766  000004                 TST     4(SP)           ;;YES, IS IT THE FIRST CHAR?
037744  001403                         BEQ     11$             ;;BRANCH IF YES
037746  016677  000002  141164         MOV     2(SP),@SWR      ;;SAVE NEW SWR
037754  062706  000006          11$:   ADD     #6,SP           ;;CLEAR UP STACK
037760  104401  001313          14$:   TYPE    ,$CRLF          ;;ECHO <CR> AND <LF>
037764  123727  001135  000001         CMPB    $INTAG,#1       ;;RE-ENABLE TTY KBD INTERRUPTS?
037772  001003                         BNE     15$             ;;BRANCH IF NOT
037774  012777  000100  141142         MOV     #100,@$TKS      ;;RE-ENABLE TTY KBD INTERRUPTS
040002  000002                 15$:    RTI                     ;;RETURN
040004  004737  036542          16$:   JSR     PC,$TYPEC       ;;ECHO CHAR
040010  021627  000060                 CMP     (SP),#60        ;;CHAR < 0?
040014  002420                         BLT     18$             ;;BRANCH IF YES
040016  021627  000067                 CMP     (SP),#67        ;;CHAR > 7?
040022  003015                         BGT     18$             ;;BRANCH IF YES
040024  042726  000060                 BIC     #60,(SP)+       ;;STRIP-OFF ASCII
040030  005766  000002                 TST     2(SP)           ;;IS THIS THE FIRST CHAR
040034  001403                         BEQ     17$             ;;BRANCH IF YES
040036  006316                         ASL     (SP)            ;;NO, SHIFT PRESENT
040040  006316                         ASL     (SP)            ;;    CHAR OVER TO MAKE
040042  006316                         ASL     (SP)            ;;    ROOM FOR NEW ONE.
040044  005266  000002          17$:   INC     2(SP)           ;;KEEP COUNT OF CHAR
```

```
040050  056616  177776                  BIS     -2(SP),(SP)     ;;SET IN NEW CHAR
040054  000707                          BR      7$              ;;GET THE NEXT ONE
040056  104401  001312          18$:    TYPE    ,$QUES          ;;TYPE ?<CR><LF>
040062  000720                          BR      20$             ;;SIMULATE CONTROL-U
                                .DSABL  LSB
                                ;;*****************************************************************
                                ;*THIS ROUTINE WILL INPUT A SINGLE CHARACTER FROM THE TTY
                                ;*CALL:
                                ;*      RDCHR                   ;;INPUT A SINGLE CHARACTER FROM THE TTY
                                ;*      RETURN HERE             ;;CHARACTER IS ON THE STACK
                                ;*                              ;;WITH PARITY BIT STRIPPED OFF
                                ;
040064  011646          $RDCHR: MOV     (SP),-(SP)      ;;PUSH DOWN THE PC
040066  016666  000004  000002  MOV     4(SP),2(SP)     ;;SAVE THE PS
040074  105777  141044          1$:     TSTB    @$TKS           ;;WAIT FOR
040100  100375                  BPL     1$              ;;A CHARACTER
040102  117766  141040  000004  MOVB    @$TKB,4(SP)     ;;READ THE TTY
040110  042766  177600  000004  BIC     #^C<177>,4(SP)  ;;GET RID OF JUNK IF ANY
040116  026627  000004  000023  CMP     4(SP),#23       ;;IS IT A CONTROL-S?
040124  001013                  BNE     3$              ;;BRANCH IF NO
040126  105777  141012          2$:     TSTB    @$TKS           ;;WAIT FOR A CHARACTER
040132  100375                  BPL     2$              ;;LOOP UNTIL ITS THERE
040134  117746  141006          MOVB    @$TKB,-(SP)     ;;GET CHARACTER
040140  042716  177600          BIC     #^C177,(SP)     ;;MAKE IT 7-BIT ASCII
040144  022627  000021          CMP     (SP)+,#21       ;;IS IT A CONTROL-Q?
040150  001366                  BNE     2$              ;;IF NOT DISCARD IT
040152  000750                  BR      1$              ;;YES, RESUME
040154  026627  000004  000140  3$:     CMP     4(SP),#140      ;;IS IT UPPER CASE?
040162  002407                  BLT     4$              ;;BRANCH IF YES
040164  026627  000004  000175  CMP     4(SP),#175      ;;IS IT A SPECIAL CHAR?
040172  003003                  BGT     4$              ;;BRANCH IF YES
040174  042766  000040  000004  BIC     #40,4(SP)       ;;MAKE IT UPPER CASE
040202  000002          4$:     RTI                     ;;GO BACK TO USER
040204     136     125     015  $CNTLU: .ASCIZ  /^U/<15><12>    ;;CONTROL ''U''
040211     136     107     015  $CNTLG: .ASCIZ  /^G/<15><12>    ;;CONTROL ''G''
040216     015     012     123  $MSWR:  .ASCIZ  <15><12>/SWR = /
040227     040     040     116  $MNEW:  .ASCIZ  /  NEW = /
```

```
                                .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.
040240  010046                  $TRAP:  MOV     R0,-(SP)        ;;SAVE R0
040242  016600  000002                  MOV     2(SP),R0        ;;GET TRAP ADDRESS
040246  005740                          TST     -(R0)           ;;BACKUP BY 2
040250  111000                          MOVB    (R0),R0         ;;GET RIGHT BYTE OF TRAP
040252  006300                          ASL     R0              ;;POSITION FOR INDEXING
040254  016000  040274                  MOV     $TRPAD(R0),R0   ;;INDEX TO TABLE
040260  000200                          RTS     R0              ;;GO TO ROUTINE
                                ;;THIS IS USE TO HANDLE THE "GETPRI" MACRO
040262  011646                  $TRAP2: MOV     (SP),-(SP)      ;;MOVE THE PC DOWN
040264  016666  000004  000002          MOV     4(SP),2(SP)     ;;MOVE THE PSW DOWN
040272  000002                          RTI                     ;;RESTORE THE PSW
                                .SBTTL  TRAP TABLE
                                ;*THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLED
                                ;*BY THE "TRAP" INSTRUCTION.
                                ;       ROUTINE
                                ;       -------
040274  040262                  $TRPAD: .WORD   $TRAP2
040276  036330                          $TYPE   ;;CALL=TYPE     TRAP+1(104401)  TTY TYPEOUT ROUTINE
040300  036706                          $TYPOC  ;;CALL=TYPOC    TRAP+2(104402)  TYPE OCTAL NUMBER (WITH LEADING ZEROS)
040302  036662                          $TYPOS  ;;CALL=TYPOS    TRAP+3(104403)  TYPE OCTAL NUMBER (NO LEADING ZEROS)
040304  036722                          $TYPON  ;;CALL=TYPON    TRAP+4(104404)  TYPE OCTAL NUMBER (AS PER LAST CALL)
040306  037110                          $TYPDS  ;;CALL=TYPDS    TRAP+5(104405)  TYPE DECIMAL NUMBER (WITH SIGN)
040310  037652                          $GTSWR  ;;CALL=GTSWR    TRAP+6(104406)  GET SOFT-SWR SETTING
040312  037602                          $CKSWR  ;;CALL=CKSWR    TRAP+7(104407)  TEST FOR CHANGE IN SOFT-SWR
040314  040064                          $RDCHR  ;;CALL=RDCHR    TRAP+10(104410) TTY TYPEIN CHARACTER ROUTINE
040316  036234                          $SAVREG ;;CALL=SAVREG   TRAP+11(104411) SAVE R0-R5 ROUTINE
040320  036272                          $RESREG ;;CALL=RESREG   TRAP+12(104412) RESTORE R0-R5 ROUTINE
5423    040322  041230                  .RSET   ;;CALL=RSETUP   TRAP+13(104413) ROUTINE TO RESET STACK AND FPS
5424            000030          $TERM=.-$TRPAD
```

```
5426                                    .SBTTL  POWER DOWN AND UP ROUTINES
                                        ;;**********************************************************************
                                        ;POWER DOWN ROUTINE
040324  012737  040502  000024  $PWRDN: MOV     #$ILLUP,@#PWRVEC ;;SET FOR FAST UP
040332  012737  000340  000026          MOV     #340,@#PWRVEC+2 ;;PRIO:7
040340  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
040342  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
040344  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
040346  010346                          MOV     R3,-(SP)        ;;PUSH R3 ON STACK
040350  010446                          MOV     R4,-(SP)        ;;PUSH R4 ON STACK
040352  010546                          MOV     R5,-(SP)        ;;PUSH R5 ON STACK
040354  017746  140560                  MOV     @SWR,-(SP)      ;;PUSH @SWR ON STACK
040360  010637  040506                  MOV     SP,$SAVR6       ;;SAVE SP
040364  012737  040376  000024          MOV     #$PWRUP,@#PWRVEC ;;SET UP VECTOR
040372  000000                          HALT
040374  000776                  -       BR      .-2             ;;HANG UP
                                        ;;**********************************************************************
                                        ;POWER UP ROUTINE
040376  012737  040502  000024  $PWRUP: MOV     #$ILLUP,@#PWRVEC ;;SET FOR FAST DOWN
040404  013706  040506                  MOV     $SAVR6,SP       ;;GET SP
040410  005037  040506                  CLR     $SAVR6          ;;WAIT LOOP FOR THE TTY
040414  005237  040506          1$:     INC     $SAVR6          ;;WAIT FOR THE INC
040420  001375                          BNE     1$              ;;OF  WORD
040422  012677  140512                  MOV     (SP)+,@SWR      ;;POP STACK INTO @SWR
040426  012605                          MOV     (SP)+,R5        ;;POP STACK INTO R5
040430  012604                          MOV     (SP)+,R4        ;;POP STACK INTO R4
040432  012603                          MOV     (SP)+,R3        ;;POP STACK INTO R3
040434  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
040436  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
040440  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
040442  012737  040324  000024          MOV     #$PWRDN,@#PWRVEC ;;SET UP THE POWER DOWN VECTOR
040450  012737  000340  000026          MOV     #340,@#PWRVEC+2 ;;PRIO:7
040456  104401                          TYPE                    ;;REPORT THE POWER FAILURE
040460  041300          $PWRMG: .WORD   POWERM                  ;;POWER FAIL MESSAGE POINTER
040462  012716                  MOV     (PC)+,(SP)              ;;RESTART AT START
040464  003606          $PWRAD: .WORD   START                   ;;RESTART ADDRESS
040466  042766  000020  000002          BIC     #20,2(SP)       ;;CLEAR "T" BIT
040474  005037  035510                  CLR     $TBIT           ;;CLEAR THE "T" BIT FLAG
040500  000002                          RTI
040502  000000          $ILLUP: HALT                            ;;THE POWER UP SEQUENCE WAS STARTED
040504  000776                  BR      .-2                     ;; BEFORE THE POWER DOWN WAS COMPLETE
040506  000000          $SAVR6: 0                               ;;PUT THE SP HERE
```

```
                                        .SBTTL  ERROR TYPE OUT ROUTINE
                                ;;**************************************************************************
                                ;;**************************************************************************
                                ;*THIS ROUTINE IS CALLED TO TYPE AN ERROR MESSAGE WHICH IS INCLUDED
                                ;*IN THE ERROR MESSAGE DATA TABLE. IT IS CALLED BY THE $ERROR ROUTINE
                                ;*OR BY FIRST SETTING $ITEMB EQUAL TO THE ERROR TABLE ITEM TO BE PRINTED
                                ;*OUT AND THEN ECECUTING A:
                                ;*              JSR     PC,ERTYPE
                                ;*
040510  104401                  ERTYPE: TYPE                                    ;TYPE A CRLF
040512  001313                          .WORD   $CRLF
040514  113737  001102  001232          MOVB    $TSTNM,$TMP0
040522  042737  177400  001232          BIC     #177400,$TMP0
040530  013737  001116  001234          MOV     $ERRPC,$TMP1                    ;GET PC OF CALL
040536  010046                          MOV     R0,-(SP)                        ;SAVE R0
040540  113700  001114                  MOVB    $ITEMB,R0                       ;GET THE ITEM NUMBER.
040544  042700  177400                  BIC     #177400,R0
040550  001005                          BNE     1$
040552  013746  001116                  MOV     $ERRPC,-(SP)                    ;IF ZERO THEN JUST
040556  104402                          TYPOC                                   ;PRINT THE PC
040560  000137  041136                  JMP     ERT5
040564  022700  000377          1$:     CMP     #377,R0
040570  001005                          BNE     20$
040572  016600  000004                  MOV     4(SP),R0
040576  011000                          MOV     (R0),R0
040600  062700  000400                  ADD     #400,R0
040604  005300                  20$:    DEC     R0                              ;OTHERWISE MAKE R0 AN
040606  006300                          ASL     R0                              ;INDEX FOR THE TABLE.
040610  006300                          ASL     R0
040612  006300                          ASL     R0
040614  062700  001442                  ADD     #$ERRTB,R0
040620  012037  040630                  MOV     (R0)+,2$                        ;PICK UP THE ADDRESS
040624  001404                          BEQ     3$                              ;OF THE EM, ERROR MESSAGE
040626  104401                          TYPE
040630  000000                  2$:     .WORD   0
040632  104401                          TYPE
040634  001313                          .WORD   $CRLF
040636  012037  040646          3$:     MOV     (R0)+,4$                        ;GET THE DH,DATA HEADER
040642  001404                          BEQ     5$
040644  104401                          TYPE
040646  000000                  4$:     .WORD   0
040650  104401                          TYPE
040652  001313                          .WORD   $CRLF
040654  010146                  5$:     MOV     R1,-(SP)                        ;SAVE R1,R2 AND R3
040656  010246                          MOV     R2,-(SP)
040660  010346                          MOV     R3,-(SP)
040662  012001                          MOV     (R0)+,R1                        ;GET THE ADDRESS OF THE
                                                                                ;DATA TABLE.
040664  001001                          BNE     6$
040666  000516                          BR      ERT4                            ;RETURN IF NO DATA.
```

ERROR TYPE OUT ROUTINE

```
                                6$:     MOV     (R0),R0                 ;GET A POINTER TO THE DATA
                                                                        ;FORMAT TABLE.
                                ERT1:   TSTB    (R0)                    ;FORMAT ZERO?
                                        BNE     7$
                                        MOV     @(R1)+,-(SP)            ;FORMAT ZERO SO TYPE
                                        TYPOC                           ;AN OCTAL NUMBER.
                                        BR      ERT2
                                7$:
                                8$:     CMPB    #2,(R0)                 ;FORMAT TWO?
                                        BNE     9$
                                        MOV     @(R1)+,R2               ;FORMAT TWO SO TYPE TWO
                                        MOV     (R2)+,-(SP)             ;OCTAL NUMBERS.
                                        TYPOC
                                        TYPE
                                        .WORD   SPACE
                                        MOV     (R2),-(SP)
                                        TYPOC
                                        BR      ERT2
                                9$:     CMPB    #3,(R0)                 ;FORMAT THREE?
                                        BNE     10$
                                        MOV     @(R1)+,R2               ;FORMAT THREE SO TYPE
                                        MOV     (R2)+,-(SP)             ;FOUR OCTAL NUMBERS.
                                        TYPOC
                                        TYPE
                                        .WORD   SPACE
                                        MOV     (R2)+,-(SP)
                                        TYPOC
                                        TYPE
                                        .WORD   SPACE
                                        MOV     (R2)+,-(SP)
                                        TYPOC
                                        TYPE
                                        .WORD   SPACE
                                        MOV     (R2),-(SP)
                                        TYPOC
                                        BR      ERT2
                                10$:    CMPB    #4,(R0)                 ;FORMAT FOUR?
                                        BNE     11$
                                        MOV     @(R1)+,-(SP)            ;FORMAR FOUR SO TYPE
                                        TYPOS                           ;AN OCTAL NUMBER
                                        .BYTE   16                      ;SUPPRESSING LEADING ZEROES.
                                        .BYTE   0
                                        BR      ERT2
                                11$:    CMPB    #5,(R0)                 ;FORMAT FIVE?
                                        BNE     13$
                                        MOV     (R1)+,12$               ;FORMAT FIVE SO TYPE AN
                                        TYPE                            ;ASCIZ STRING.
```

```
5541 041032  000000                12$:   .WORD   0
5542 041034  000427                       BR      ERT3
5543
5544 041036  122710  000011        13$:   CMPB    #11,(R0)                ;FORMAT ELEVEN?
5545 041042  001005                       BNE     15$
5546 041044  013137  041052               MOV     @(R1)+,14$              ;FORMAT ELEVEN SO PICK
5547 041050  104401                       TYPE                            ;A POINTER TO AN ASCIZ
5548 041052  000000                14$:   .WORD   0                       ;STRING.
5549 041054  000417                       BR      ERT3
5550
5551 041056  122710  000012        15$:   CMPB    #12,(R0)                ;FORMAT TWELVE?
5552 041062  001011                       BNE     17$
5553
5554 041064  013102                       MOV     @(R1)+,R2               ;FORMAT TWELVE SO TYPE
5555 041066  012703  000006               MOV     #6,R3                   ;TYPE SIX OCTAL NUMBERS
5556 041072  012246                16$:   MOV     (R2)+,-(SP)
5557 041074  104402                       TYPOC
5558 041076  104401                       TYPE
5559 041100  041350                       .WORD   SPACE
5560 041102  077305                       SOB     R3,16$
5561 041104  000401                       BR      ERT2
5562
5563 041106  000000                17$:   HALT                            ;UNDEFINED FORMAT FOR DATA?????
5564
5565 041110  104401                FRT2:  TYPE                            ;PRINT A TAB AFTER TYPING
5566 041112  041346                       .WORD   $TAB                    ;AN DATA TABLE ENTRY
5567                                                                       ;OF ALL FORMATS EXCEPT
5568                                                                       ;ASCIZ, FORMATS 5 OR 11
5569
5570 041114  005200                ERT3:  INC     R0                      ;POINT TO THE NEXT FORMAT
5571 041116  005711                       TST     (R1)                    ;END OF DATA TABLE.
5572 041120  001401                       BEQ     ERT4
5573 041122  000663                       BR      ERT1
5574
5575 041124  104401                ERT4:  TYPE                            ;DONE.
5576 041126  001313                       .WORD   $CRLF
5577 041130  012603                       MOV     (SP)+,R3                ;RESTORE R1,R2 AND R3
5578 041132  012602                       MOV     (SP)+,R2
5579 041134  012601                       MOV     (SP)+,R1
5580 041136  012600                ERT5:  MOV     (SP)+,R0                ;RESTORE R0.
5581 041140  000207                       RTS     PC                      ;AND RETURN.
```

FPP SPURIOUS TRAP TO 244 HANDLER

```
                                        .SBTTL  FPP SPURIOUS TRAP TO 244 HANDLER
                                ;;**************************************************************************
                                ;;**************************************************************************
                                ;*THIS ROUTINE HANDLES UNEXPECTED TRAPS TO THE FPP TRAP VECTOR AT 244.
                                ;*THE LAST FPP INSTRUCTION EXECUTED AND ITS ADDRESS HAS BEEN RECORDED
                                ;*THESE ALONG WITH THE FEC, FPS AND PC OF TRAP ARE REPORTED.
                                ;*
041142  011637  001236          FPSPUR: MOV     (SP),$TMP2              ;SAVE PC OF TRAP.
041146  022626                          CMP     (SP)+,(SP)+             ;RESTORE SP.
041150  170200                          STFPS   R0                      ;GET FPS
041152  010037  001240                  MOV     R0,$TMP3
041156  170300                          STST    R0                      ;GET FEC
041160  010037  001242                  MOV     R0,$TMP4
041164  104211                  1$:     ERROR   +211
041166  104413                          RSETUP                          ;GO INITIALIZE THE FPS AND STACK; AND
                                                                        ;SEE IF THE USER HAS EXPRESSED
                                                                        ;THE DESIRE TO CHANGE THE SOFTWARE
                                                                        ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
                                                                        ;THE USER TYPED CONTROL G?).
041170  000137  035222                  JMP     $EOP
```

```
                                        .SBTTL  CPU SPURIOUS TRAP TO 4 HANDLER
                                ;;**************************************************************
                                ;;**************************************************************
                                ;*THIS ROUTINE REPORTS UNEXPECTED CPU TRAPS TO VECTOR 4.
                                ;*
5601 041174  011637  001236     CPSPUR: MOV     (SP),$TMP2              ;SAVE PC OF TRAP.
5602 041200  022626             1$:     CMP     (SP)+,(SP)+
5603 041202  104212                     ERROR   +212
5604 041204  104413                     RSETUP                          ;GO INITIALIZE THE FPS AND STACK; AND
                                                                        ;SEE IF THE USER HAS EXPRESSED
                                                                        ;THE DESIRE TO CHANGE THE SOFTWARE
                                                                        ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
                                                                        ;THE USER TYPED CONTROL G?).
5605 041206  000137  035222             JMP     $EOP
```

```
                                .SBTTL  CPU SPURIOUS TRAP TO 10 HANDLER
                        ;;**********************************************************************
                        ;;**********************************************************************
                        ;*THIS ROUTINE REPORTS UNEXPECTED CPU TRAPS TO VECTOR 10.
                        ;*
041212  011637  001236  CPTWO:  MOV     (SP),$TMP2              ;SAVE PC OF TRAP.
041216  022626          CMP     (SP)+,(SP)+
041220  104213          1$:     ERROR   +213
041222  104413                  RSETUP                          ;GO INITIALIZE THE FPS AND STACK; AND
                                                                ;SEE IF THE USER HAS EXPRESSED
                                                                ;THE DESIRE TO CHANGE THE SOFTWARE
                                                                ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
                                                                ;THE USER TYPED CONTROL G?).
041224  000137  035222          JMP     $EOP
```

```
                                        .SBTTL  FLAG RESET AND CONSOLE TEST ROUTINE
                                ;;**************************************************************
                                ;;**************************************************************
                                ;*THIS ROUTINE WILL BE CALLED AT THE END OF EACH TEST TO
                                ;*RESET THE STACK, CLEAR THE FPS AND SEE IF THE USER HAS TYPED
                                ;* CONTROL G ON THE TERMINAL. IF THE USER HAS TYPED CONTROL G AND
                                ;*THERE IS NO PHYSICAL CONSOLE SWITCH REGISTER THEN THE CONTENTS
                                ;*OF THE SOFTWARE SWITCH REGISTER WILL BE TYPED IN OCTAL ON THE
                                ;*TELETYPE AND THE USER CAN MODIFY IT.
                                ;*
041230  023727  001140  177570  .RSET:  CMP     SWR,#177570             ;SEE IF THERE IS A PHYSICAL
                                                                        ;CONSOLE SWITCH REGISTER.
041236  001001                          BNE     1$                      ;BRANCH IF NO.
041240  104407                          CKSWR                           ;OTHERWISE TYPE THE CONTENTS
                                                                        ;OF THE PROGRAM VIRTUAL SWITCH REGISTER
                                                                        ;AND GIVE THE USER A CHANCE TO
                                                                        ;MODIFY IT.
041242  012737  041142  000244  1$:     MOV     #FPSPUR,FPVECT
041250  012737  041174  000004          MOV     #CPSPUR,ERRVECT
041256  012737  041212  000010          MOV     #CPTWO,10
041264  011600                          MOV     (SP),R0                 ;SAVE RETURN ADDRESS.
041266  012706  001100                  MOV     #STACK,SP               ;RESET THE STACK POINTER.
041272  005004                          CLR     R4                      ;CLEAR THE FPS.
041274  170104                          LDFPS   R4
041276  000110                          JMP     (R0)                    ;RETURN.
```

SPECIAL MESSAGES

```
                                                .SBTTL  SPECIAL MESSAGES
041300  200  120  117  POWERM: .ASCIZ  <CRLF>'POWER FAILURE. PROGRAM RESTARTING.'<CRLF>
041345  000       NULL:   .BYTE   0
041346  011  000  $TAB:   .ASCIZ  <TAB>
041350  040  040  000  SPACE:  .ASCIZ  ' '
041353  200  120  103  LFIEX1: .ASCIZ  <CRLF>'PC OF LAST FPP INSTRUCTION EXECUTED: '<TAB>
041423  200  114  101  LFIEX2: .ASCIZ  <CRLF>'LAST FPP INSTRUCTION EXECUTED: '<TAB>
041465  200  106  114  FPSMS:  .ASCIZ  <CRLF>'FLOATING POINT STATUS REGISTER:  '
041531  200  106  105  FECMS:  .ASCIZ  <CRLF>'FEC:  '
041542  124  110  105  $THE:   .ASCIZ  'THE '
041547  011  040  111  NOOP1:  .ASCIZ  <TAB>' INSTRUCTION FAILED.'<CRLF>
041576  105  111  124  NOOP15: .ASCII  'EITHER A BAD CONSTANT WAS GENERATED OR MICROPROGRAM FLOW WENT'
041673  200  106  122  NOOP2:  .ASCIZ  <CRLF>'FROM STATE '
041710  124  117  040  NOOP3:  .ASCIZ  'TO STATE '
041722  200  111  116  NOOP4:  .ASCIZ  <CRLF>'INSTEAD OF '
041737  200  124  110  NOOP5:  .ASCIZ  <CRLF>'THEREBY EXECUTING A '
041765  011  040  111  NOOP6:  .ASCIZ  <TAB>' INSTEAD OF A '
042005  011  040  111  NOOP7:  .ASCIZ  <TAB>' INSTRUCTION.'<CRLF>
042025  040  040  124  NOOP10: .ASCII  '  TEST.'<TAB>'PC OF CALL.'<TAB>'PC OF ERROR.'<TAB>
042066  107  117  124       .ASCIZ  'GOT FPS. EXPECTED FPS.'<CRLF>
042116  101  040  102  NOOP11: .ASCIZ  'A BAD CONSTANT MAY HAVE BEEN USED.'<CRLF>
042162  011  114  104  LFPS1:  .ASCIZ  <TAB>'LDFPS'<TAB>'REG'
042175  011  114  104  LD1:    .ASCIZ  <TAB>'LDD'<TAB>'(REG),A'<TAB>'//FSRC#0//'
042225  011  114  104  LD2:    .ASCIZ  <TAB>'LDD'<TAB>'A,A'
042236  011  123  124  STFS1:  .ASCIZ  <TAB>'STFPS'<TAB>'REG'
042251  011  123  124  ST1:    .ASCIZ  <TAB>'STD'<TAB>'A,(REG)'
042266  011  123  124  ST2:    .ASCIZ  <TAB>'STD'<TAB>'A,A'
042277  011  103  106  CFCC1:  .ASCIZ  <TAB>'CFCC'
042305  011  123  105  SETF1:  .ASCIZ  <TAB>'SETF'
042313  011  123  105  SETD1:  .ASCIZ  <TAB>'SETD'
042321  011  123  105  SETI1:  .ASCIZ  <TAB>'SETI'
042327  011  123  105  SETL1:  .ASCIZ  <TAB>'SETL'
042335  011  111  114  ILL1:   .ASCIZ  <TAB>'ILLEGAL FPP INSTRUCTION'
042366  011  123  124  STST1:  .ASCIZ  <TAB>'STST'<TAB>'REG'
042400  011  111  114  ILL2:   .ASCIZ  <TAB>'ILLEGAL FPP INSTRUCTION (FID=1)'
042441  040  040  124  ILLMS:  .ASCIZ  '  TEST.'<TAB>'PC OF CALL.'<TAB>'PC OF TRAP.'<TAB>'FPS.'<CRLF>
042507  105  130  120  MS1:    .ASCIZ  'EXPECTED '
042521  107  117  124  MS2:    .ASCIZ  'GOT '
042526  103  117  116  MS3:    .ASCIZ  'CONTENTS OF LOCATIONS '
042555  040  124  110  MS4:    .ASCIZ  ' THROUGH '
042567  106  101  111  MS5:    .ASCIZ  'FAILURE IN THE MICROPROGRAM FLOW.'
042631  103  117  116  MS6:    .ASCIZ  'CONTROL WENT '
042647  106  122  117  MS7:    .ASCIZ  'FROM STATE '
042663  040  124  117  MS10:   .ASCIZ  ' TO STATE '
042676  102  125  124  MS11:   .ASCIZ  'BUT SHOULD HAVE GONE'
042723  103  117  116  MS12:   .ASCIZ  'CONTROL FLOW SHOULD HAVE GONE'
042761  102  125  124  MS13:   .ASCIZ  'BUT DID NOT.'
042776  040  040  124  MS14:   .ASCII  '  TEST.'<TAB>'PC OF CALL.'<TAB>'PC OF ERROR.'<TAB>
043037  107  117  124       .ASCIZ  'GOT PC.'<TAB>'EXPECTED PC.'
043064  111  116  123  MS15:   .ASCIZ  'INSTRUCTION TESTED: '
043111  040  117  122  MS16:   .ASCIZ  ' OR '
043116  124  105  123  MS17:   .ASCIZ  'TESTING ACCUMULATOR '
043143  132  105  122  MNUM0:  .ASCIZ  'ZERO '
043151  117  116  105  MNUM1:  .ASCIZ  'ONE '
043156  124  127  117  MNUM2:  .ASCIZ  'TWO '
043163  124  110  122  MNUM3:  .ASCIZ  'THREE '
043172  106  117  125  MNUM4:  .ASCIZ  'FOUR '
```

SPECIAL MESSAGES

```
 5696 043200  106  111  126  MNUM5:  .ASCIZ  'FIVE '
 5697 043206  040  040  124  MS20:   .ASCIZ  '  TEST.'<TAB>'PC OF CALL.'<TAB>'PC OF ERROR.'
 5698 043247  104  101  124  MS21:   .ASCIZ  'DATA (FLOATING POINT NUMBER): '
 5699 043306  114  117  107  MS22:   .ASCIZ  'LOGICAL AND OF FAILING '
 5700 043336  114  117  107  MS23:   .ASCIZ  'LOGICAL OR OF FAILING '
 5701 043365  040  040  124  MS24:   .ASCII  '  TEST.'<TAB>'PC OF CALL.'<TAB>'PC OF ERRORS.'<TAB>
 5702 043427  116  125  115          .ASCIZ  'NUMBER OF ERRORS(OCTAL).'
 5703 043460  105  130  120  MS25:   .ASCIZ  'EXPECTED DATA IN '
 5704 043502  107  117  124  MS26:   .ASCIZ  'GOT DATA IN '
 5705 043517  200  101  103  MS27:   .ASCIZ  <CRLF>'AC0= '
 5706 043526  200  101  103  MS30:   .ASCIZ  <CRLF>'AC1= '
 5707 043535  200  101  103  MS31:   .ASCIZ  <CRLF>'AC2= '
 5708 043544  200  101  103  MS32:   .ASCIZ  <CRLF>'AC3= '
 5709 043553  200  101  103  MS33:   .ASCIZ  <CRLF>'AC4= '
 5710 043562  200  101  103  MS34:   .ASCIZ  <CRLF>'AC5= '
 5711 043571  123  105  124  MS35:   .ASCIZ  'SET '
 5712 043576  103  114  105  MS36:   .ASCIZ  'CLEAR '
 5713 043605  114  117  101  MS37:   .ASCIZ  'LOADED DATA: '
 5714 043623  122  105  101  MS40:   .ASCIZ  'READ DATA: '
 5715 043637  105  130  120  MS415:  .ASCIZ  'EXPECTED DATA: '
 5716 043657  104  101  124  MS41:   .ASCIZ  'DATA IN (R0) FSRC: '
 5717 043703  104  101  124  MS42:   .ASCIZ  'DATA IN AC0: '
 5718 043721  107  117  124  MS43:   .ASCIZ  'GOT RESULT: '
 5719 043736  105  130  120  MS44:   .ASCIZ  'EXPECTED RESULT: '
```

ERROR MESSAGES

```
5720                                            .SBTTL  ERROR MESSAGES
5721 043760     114     104     106  EM1:      .ASCIZ  'LDFPS AND STFPS TEST FAILED.'
5722 044015     114     104     106  EM2:      .ASCIZ  'LDFPS AND STFPS TEST ERROR SUMMARY.'
5723 044061     103     106     103  EM3:      .ASCIZ  'CFCC TRANSFERED BAD DATA TO THE PSW.'
5724 044126     103     106     103  EM4:      .ASCIZ  'CFCC MODIFIED THE FPS REGISTER.'
5725 044166     125     116     105  EM5:      .ASCIZ  'UNEXPECTED FPP TRAP TO 244.'
5726 044222     125     116     105  EM6:      .ASCIZ  'UNEXPECTED CPU TRAP TO 4.'
5727 044254     125     116     105  EM7:      .ASCIZ  'UNEXPECTED CPU TRAP TO 10.'
5728         044166                  EM10=EM5
5729 044307     125     116     101  EM11:     .ASCIZ  'UNABLE TO DECODE FPP INSTRUCTION. TRAPPED TO 10.'
5730         000000                  EM12=0
5731         000000                  EM13=0
5732 044370     114     104     106  EM14:     .ASCII  'LDFPS R0 FAILED IN THE FSRC FLOWS.'
5733 044432     040     124     122              .ASCII  ' TRAPPED TO 4.'
5734 044450     200     104     111              .ASCIZ  <CRLF>'DID NOT GO FROM STATE 400 TO 670.'
5735 044513     123     124     106  EM15:     .ASCII  'STFPS R1 FAILED IN THE FDST FLOWS.'
5736 044555     040     124     122              .ASCII  ' TRAPPED TO 4.'
5737 044573     200     104     111              .ASCIZ  <CRLF>'DID NOT GO FROM STATE 634 TO 710.'
5738 044636     101     116     040  EM16:     .ASCIZ  'AN ILLEGAL FPP INSTRUCTION DID NOT TRAP.'
5739 044707     101     116     040  EM17:     .ASCII  'AN ILLEGAL FPP INSTRUCTION'
5740 044741     200     124     122              .ASCII  <CRLF>'TRAPPED TO 244, BUT FAILED TO SET '
5741 045004     124     110     105              .ASCII  'THE FPS CORRECTLY.'<CRLF>'EITHER A BAD CONSTANT '
5742 045055     127     101     123              .ASCIZ  'WAS GENERATED OR THE ALU LOGICAL OR FUNCTION FAILED.'
5743 045142     101     116     040  EM20:     .ASCII  'AN ILLEGAL FPP INSTRUCTION'
5744 045174     040     124     122              .ASCII  ' TRAPPED TO 244, BUT A SUBSEQUENT '
5745 045236     040     123     124              .ASCII  ' STST'<CRLF>
5746 045244     106     101     111              .ASCIZ  'FAILED TO PICK UP THE CORRECT FEC CODE = 2.'
5747 045320     123     124     123  EM21:     .ASCII  'STST R4 FAILED IN THE DESTINATION FLOWS.'
5748 045370     040     124     122              .ASCII  ' TRAPPED TO 4.'<CRLF>
5749 045407     104     111     104              .ASCIZ  'DID NOT GO FROM STATE 636 TO 710.'
5750 045451     101     116     040  EM22:     .ASCII  'AN ILLEGAL FPP INSTRUCTION,'
5751 045504     127     111     124              .ASCIZ  'WITH INTERRUPTS DISABLED,'
5752         045451                  EM23=EM22
5753         045451                  EM24=EM22
5754 045536     123     117     125  EM25:     .ASCII  'SOURCE LOCATIONS MODIFIED BY, LDD.'
5755 045600     200     101     040              .ASCIZ  <CRLF>'A DATO WAS PERFORMED INSTEAD OF A DATI.'
5756 045651     114     104     104  EM26:     .ASCII  'LDD (R0),AC0 FAILED.'<CRLF>
5757 045676     122     060     040              .ASCIZ  'R0 WAS MODIFIED.'
5758         045651                  EM27=EM26
5759 045717     124     110     105  EM30:     .ASCII  'THE PC WAS BAD AFTER  '
5760 045745     101     116     040              .ASCIZ  'AN FPP INSTRUCTION.'
5761 045771     123     124     104  EM31:     .ASCII  'STD AC0,(R0) FAILED.'<CRLF>
5762 046016     122     060     040              .ASCIZ  'R0 WAS MODIFIED.'
5763         045771                  EM32=EM31
5764 046037     123     124     104  EM33:     .ASCII  'STD AC0,(R0) FAILED.'<CRLF>
5765 046064     117     125     124              .ASCIZ  'OUTPUT BAD.'
5766 046100     123     124     104  EM34:     .ASCII  'STD AC0,(R0) FAILED IN THE FDST FLOWS.'
5767 046146     200     124     110              .ASCIZ  <CRLF>'THE (BUT GR7) FORK FAILED.'
5768 046202     114     104     104  EM35:     .ASCII  'LDD (R0),AC0 FAILED IN THE FSRC FLOWS.'
5769 046250     200     124     110              .ASCIZ  <CRLF>'THE (BUT GR7) FORK FAILED.'
5770 046304     123     124     104  EM36:     .ASCII  'STD AC0,(R0) FAILED IN THE FDST FLOWS.'
5771 046352     200     124     110              .ASCIZ  <CRLF>'THE (BUT FD) FORK FAILED.'
5772 046405     114     104     104  EM37:     .ASCII  'LDD (R0),AC0 FAILED IN THE FSRC FLOWS.'
5773 046453     200     124     110              .ASCIZ  <CRLF>'THE (BUT FD) FORK FAILED.'
5774 046506     114     104     104  EM40:     .ASCII  'LDD (R0),AC0 OR THE STD AC0,(R0) FAILED.'
5775 046556     200     102     101              .ASCIZ  <CRLF>'BAD DATA WAS DETECTED AFTER A SEQUENCE OF THE TWO INSTRUCTIONS.'
5776 046657     106     120     123  EM41:     .ASCIZ  'FPS BAD AFTER EXECUTION OF: '
```

ERROR MESSAGES

```
 5781 046714                            EM42:
      046714   114   104   104                 .ASCII  /LDD (R0),AC0 FAILED IN THE FSRC FLOWS./<CRLF>
      046763   124   110   105                 .ASCIZ  /THE (BUT FSRC) FORK FAILED. TRAPPED TO 4./
 5782 047035                            EM43:
      047035   123   124   104                 .ASCII  /STD AC0,(R0) FAILED IN THE FDST FLOWS./<CRLF>
      047104   124   110   105                 .ASCIZ  /THE (BUT FDST) FORK FAILED. TRAPPED TO 4./
 5783 047156   106   120   120  EM44:          .ASCIZ  'FPP ACCUMULATORS DATA TEST FAILED.'
 5784        047156              EM45=EM44
 5785 047221   106   120   120  EM46:          .ASCIZ  'FPP ACCUMULATORS DUAL ADDRESSING TEST FAILED.'
 5790 047277                            EM47:
      047277   114   104   040                 .ASCII  /LD AC1,AC0 FAILED IN THE FSRC FLOWS./
      047343   124   110   105                 .ASCIZ  /THE (BUT FSRC) FORK FAILED. TRAPPED TO 4./
 5791 047415   114   104   040  EM50:          .ASCII  'LD AC1,AC0 FAILED IN THE FSRC FLOWS.'
 5792 047461   124   110   105                 .ASCIZ  'THE (BUT FD) FORK FAILED.'
 5793 047513   114   104   040  EM51:          .ASCIZ  'LD AC1,AC0 TRANSFERRED BAD DATA.'
 5803 047554                            EM52:
      047554   114   104   104                 .ASCII  /LDD (R0)+,AC0 FAILED IN THE FSRC FLOWS./
      047623   124   110   105                 .ASCIZ  /THE (BUT FSRC) FORK FAILED. TRAPPED TO 4./
 5804 047675                            EM53:
      047675   114   104   104                 .ASCII  /LDD (R0)+,AC0 FAILED IN THE FSRC FLOWS./
      047744   200   122   060                 .ASCII  <CRLF>'R0 WAS BAD.'<CRLF>
      047761   105   111   124                 .ASCII  'EITHER A BAD CONSTANT WAS GENERATED OR'<CRLF>
      050030   104   111   104                 .ASCIZ  \DID NOT GO FROM STATE 627 TO 322.\
 5805 050072                            EM54:
      050072   114   104   104                 .ASCIZ  /LDD (R0)+,AC0 TRANSFERRED BAD DATA./
 5806 050136                            EM55:
      050136   114   104   104                 .ASCII  /LDD -(R0),AC0 FAILED IN THE FSRC FLOWS./
      050205   124   110   105                 .ASCIZ  /THE (BUT FSRC) FORK FAILED. TRAPPED TO 4./
 5807 050257                            EM56:
      050257   114   104   104                 .ASCII  /LDD -(R0),AC0 FAILED IN THE FSRC FLOWS./
      050326   200   122   060                 .ASCII  <CRLF>'R0 WAS BAD.'<CRLF>
      050343   105   111   124                 .ASCII  'EITHER A BAD CONSTANT WAS GENERATED OR'<CRLF>
      050412   104   111   104                 .ASCIZ  \DID NOT GO FROM STATE 627 TO 324.\
 5808 050454                            EM57:
      050454   114   104   104                 .ASCIZ  /LDD -(R0),AC0 TRANSFERRED BAD DATA./
 5809 050520                            EM60:
      050520   114   104   106                 .ASCII  /LDF (R0)+,AC0 FAILED IN THE FSRC FLOWS./
      050567   200   122   060                 .ASCII  <CRLF>'R0 WAS BAD.'<CRLF>
      050604   105   111   124                 .ASCII  'EITHER A BAD CONSTANT WAS GENERATED OR'<CRLF>
      050653   104   111   104                 .ASCIZ  \DID NOT GO FROM STATE 627 TO 322.\
 5810 050715                            EM61:
      050715   114   104   106                 .ASCIZ  /LDF (R0)+,AC0 TRANSFERRED BAD DATA./
 5811 050761   114   104   106  EM62:          .ASCII  'LDF (R0)+,AC0 FAILED IN THE FSRC FLOWS.'
 5812 051030   200   124   110                 .ASCII  <CRLF>'THE (BUT FD) FORK FAILED.'<CRLF>
 5813 051063   127   105   116                 .ASCII  'WENT FROM STATE 441 TO 077,'<CRLF>
 5814 051117   111   116   123                 .ASCIZ  'INSTEAD OF FROM 441 TO 076.'
 5815 051153   114   104   104  EM63:          .ASCII  'LDD #NUM,AC0 FAILED IN THE FSRC FLOWS.'
 5816 051221   200   124   110                 .ASCII  <CRLF>'THE (BUT GR7) FORK FAILED.'<CRLF>
 5817 051255   127   105   116                 .ASCII  'WENT FROM STATE 207 TO 174,'<CRLF>
 5818 051311   111   116   123                 .ASCIZ  'INSTEAD OF FROM 207 TO 176.'
 5819 051345   114   104   104  EM64:          .ASCII  'LDD #NUM,AC0 FAILED IN THE FSRC FLOWS.'
 5820 051413   200   101   040                 .ASCIZ  <CRLF>'A BAD CONSTANT WAS USED WHEN THE PC WAS INCREMENTED.'
 5821        051345              EM65=EM64
 5822 051501                            EM66:
      051501   114   104   104                 .ASCIZ  /LDD #NUM,AC0 TRANSFERRED BAD DATA./
 5844 051544                            EM67:
      051544   114   104   104                 .ASCII  'LDD @(R0)+,AC0 FAILED IN THE FSRC FLOWS.'
```

```
                051614  200  124  110          .ASCII  <CRLF>'THE (BUT FSRC) FORK FAILED. TRAPPED TO 4.'
                051666  200  127  105          .ASCII  <CRLF>\WENT FROM STATE 627 TO EITHER 326 OR 326,\
                051740  200  111  116          .ASCIZ  <CRLF>\INSTEAD OF FROM 627 TO 323.\
5845            051775                  EM70:
                051775  114  104  104          .ASCII  'LDD @(R0)+,AC0 FAILED IN THE FSRC FLOWS.'
                052045  200  124  110          .ASCIZ  <CRLF>'THE (BUT FSRC) FORK FAILED. TRAPPED TO 4.'
5846            052120                  EM71:
                052120  114  104  104          .ASCII  'LDD @(R0)+,AC0 FAILED IN THE FSRC FLOWS.'
                052170  124  110  105          .ASCIZ  'THE (BUT FD) FORK FAILED.'
5847            052222                  EM72:
                052222  114  104  104          .ASCII  'LDD @(R0)+,AC0'<CRLF>
                052241  106  101  111          .ASCIZ  'FAILED TO INCREMENT R0 BY 2.'
5848            052276                  EM73:
                052276  114  104  104          .ASCIZ  'LDD @(R0)+,AC0 LOADED BAD DATA.'
5849            052336                  EM74:
                052336  114  104  104          .ASCII  'LDD @-(R0),AC0 FAILED IN THE FSRC FLOWS.'
                052406  200  124  110          .ASCII  <CRLF>'THE (BUT FSRC) FORK FAILED. TRAPPED TO 4.'
                052460  200  127  105          .ASCII  <CRLF>\WENT FROM STATE 627 TO EITHER 326 OR 326,\
                052532  200  111  116          .ASCIZ  <CRLF>\INSTEAD OF FROM 627 TO 325.\
5850            052567                  EM75:
                052567  114  104  104          .ASCII  'LDD @-(R0),AC0 FAILED IN THE FSRC FLOWS.'
                052637  200  124  110          .ASCIZ  <CRLF>'THE (BUT FSRC) FORK FAILED. TRAPPED TO 4.'
5851            052712                  EM76:
                052712  114  104  104          .ASCII  'LDD @-(R0),AC0 FAILED IN THE FSRC FLOWS.'
                052762  124  110  105          .ASCIZ  'THE (BUT FD) FORK FAILED.'
5852            053014                  EM77:
                053014  114  104  104          .ASCII  'LDD @-(R0),AC0'<CRLF>
                053033  106  101  111          .ASCIZ  'FAILED TO DECREMENT R0 BY 2.'
5853            053070                  EM100:
                053070  114  104  104          .ASCIZ  'LDD @-(R0),AC0 LOADED BAD DATA.'
5854            053130                  EM101:
                053130  114  104  104          .ASCII  'LDD NUM(R0),AC0 FAILED IN THE FSRC FLOWS.'
                053201  200  124  110          .ASCIZ  <CRLF>'THE (BUT FSRC) FORK FAILED. TRAPPED TO 4.'
5855            053254                  EM102:
                053254  114  104  104          .ASCII  'LDD NUM(R0),AC0'<CRLF>
                053274  106  101  111          .ASCIZ  'FAILED TO AFFECT R0 BY 2.'
5856            053326                  EM103:
                053326  114  104  104          .ASCII  'LDD NUM(R0),AC0 FAILED IN THE FSRC FLOWS.'
                053377  124  110  105          .ASCIZ  'THE (BUT FD) FORK FAILED.'
5857            053431                  EM104:
                053431  114  104  104          .ASCIZ  'LDD NUM(R0),AC0 LOADED BAD DATA.'
5858            053472                  EM105:
                053472  114  104  104          .ASCII  'LDD @NUM(R0),AC0 FAILED IN THE FSRC FLOWS.'
                053544  200  124  110          .ASCIZ  <CRLF>'THE (BUT FSRC) FORK FAILED. TRAPPED TO 4.'
5859            053617                  EM106:
                053617  114  104  104          .ASCII  'LDD @NUM(R0),AC0'<CRLF>
                053640  106  101  111          .ASCIZ  'FAILED TO AFFECT R0 BY 2.'
5860            053672                  EM107:
                053672  114  104  104          .ASCII  'LDD @NUM(R0),AC0 FAILED IN THE FSRC FLOWS.'
                053744  124  110  105          .ASCIZ  'THE (BUT FD) FORK FAILED.'
5861            053776                  EM110:
                053776  114  104  104          .ASCIZ  'LDD @NUM(R0),AC0 LOADED BAD DATA.'
5877            054040                  EM111:
                054040  114  104  104          .ASCII  /LDD AC7,AC0 FAILED TO TRAP TO 244./
                054102  200  101  103          .ASCIZ  <CRLF>/AC7 IS AN ILLEGAL ACCUMULATOR./
5878                 054040             EM112=EM111
5879            054142                  EM113:
```

```
        054142      114      104      104          .ASCII  /LDD AC6,AC0 FAILED TO TRAP TO 244./
        054204      200      101      103          .ASCIZ  <CRLF>/AC6 IS AN ILLEGAL ACCUMULATOR./
 5880             054142                     EM114=EM113
 5881             054040                     EM115=EM111
 5882             054142                     EM116=EM113
 5883 054244                                 EM117:
        054244      125      123      105          .ASCII  'USE OF AN ILLEGAL ACCUMULATOR WITH FSRC MODE ZERO,'
        054326      200      124      122          .ASCIZ  <CRLF>'TRAPPED BUT FAILED TO SET FPS CORRECTLY.'
 5884 054400                                 EM120:
        054400      125      123      105          .ASCII  'USE OF AN ILLEGAL ACCUMULATOR WITH FSRC MODE ZERO,'
        054462      200      124      122          .ASCIZ  <CRLF>'TRAPPED BUT FAILED TO SET FEC CORRECTLY.'
 5885 054534      123      124      040  EM121:  .ASCII  'ST AC0,AC1 FAILED IN THE FDST FLOWS.'
 5886 054600      200      124      110          .ASCIZ  <CRLF>'THE (BUT FDST) FORK FAILED. TRAPPED TO 4.'
 5887 054653      123      124      040  EM122:  .ASCII  'ST AC0,AC1 FAILED IN THE FDST FLOWS.'
 5888 054717      200      124      110          .ASCIZ  <CRLF>'THE (BUT FD) FORK FAILED.'
 5889 054752      123      124      040  EM123:  .ASCIZ  'ST AC0,AC1 TRANSFERRED BAD DATA.'
 5890 055013                                 EM124:
        055013      106      120      123          .ASCII  'FPS BAD AFTER LDD (R0),AC0.'
        055046      200      124      110          .ASCIZ  <CRLF>\THE (BUT EZBT Y8) FORK FAILED.\
 5891 055106                                 EM125:
        055106      106      120      123          .ASCII  'FPS BAD AFTER LDD (R0),AC0.'
        055141      200      124      110          .ASCIZ  <CRLF>\THE (BUT ENBT) FORK FAILED.\
 5892 055176      114      104      104  EM126:  .ASCII  'LDD (R0),AC0 TRAPPED TO 244.'
 5893 055232      040      106      123          .ASCII  ' FSRC= -0 AND FIUV= 0.'<CRLF>
 5894 055261      124      110      105          .ASCII  'THE (BUT FIUV) FORK FAILED.'
 5895 055314      200      127      105          .ASCII  <CRLF>'WENT FROM STATE 256 TO 354,'
 5896 055350      200      111      116          .ASCIZ  <CRLF>'INSTEAD OF FROM 256 TO 254.'
 5897 055405      114      104      104  EM127:  .ASCII  'LDD (R0),AC0 FAILED TO TRAP TO 244,'
 5898 055450      040      106      123          .ASCII  ' FSRC= -0, FIUV= 1.'
 5899 055473      200      124      110          .ASCII  <CRLF>'THE (BUT FIUV) FOR FAILED.'<CRLF>
 5900 055527      127      105      116          .ASCII  'WENT FROM STATE 256 TO 254,'
 5901 055562      200      111      116          .ASCIZ  <CRLF>'INSTEAD OF FROM 256 THE 354.'
 5902 055620      114      104      104  EM130:  .ASCII  'LDD (R0),AC0 TRAPPED TO 244,'
 5903 055654      106      123      122          .ASCII  'FSRC= -0, FIUV= 1.'<CRLF>
 5904 055677      102      125      124          .ASCIZ  'BUT FEC WAS BAD.'
 5905 055720                                 EM131:
        055720      114      104      103          .ASCIZ  /LDCFD (R0),AC0 LOADED BAD DATA./
 5906 055760                                 EM132:
        055760      114      104      103          .ASCIZ  /LDCDF (R0),AC0 LOADED BAD DATA./
 5947 056020                                 EM133:
        056020      101      104      104          .ASCIZ  /ADDD (R0),AC0 WITH (R0)=AC0=0 /
 5948 056057                                 EM134:
        056057      101      104      104          .ASCIZ  /ADDF (R0),AC0 WITH (R0)=AC0=0 /
 5949 056116                                 EM135:
        056116      123      125      102          .ASCIZ  /SUBD (R0),AC0 WITH (R0)=AC0=0 /
 5950 056155                                 EM136:
        056155      123      125      102          .ASCIZ  /SUBF (R0),AC0 WITH (R0)=AC0=0 /
 5951             056020                     EM137=EM133
 5952             056057                     EM140=EM134
 5953             056116                     EM141=EM135
 5954             056155                     EM142=EM136
 5955 056214                                 EM143:
        056214      101      104      104          .ASCIZ  /ADDD (R0),AC0 WITH (R0)=0 /
 5956 056247                                 EM144:
        056247      123      125      102          .ASCIZ  /SUBD (R0),AC0 WITH (R0)=0 /
 5957             056214                     EM145=EM143
 5958             056247                     EM146=EM144
```

ERROR MESSAGES

```
                                        EM147:
056302                                          .ASCIZ  /SUBD (R0),AC0 WITH AC0=0 /
056302   123   125   102
                056302                  EM150=EM147
                056302                  EM151=EM147
056334                                  EM152:
056334   101   104   104                          .ASCIZ  /ADDD (R0),AC0 WITH AC0=0 /
                056334                  EM153=EM152
056366                                  EM154:
056366   101   116   040                          .ASCII  'AN OVERFLOW ERROR OCCURRED ON ADD'<CRLF>
056430   103   101   125                          .ASCII  'CAUSING A TRAP TO 244.'
056456   200   050   102                          .ASCII  <CRLF>'(BUT EZBT Y9 Y8) FORK IN STATE 420 OF OVER\UNDER FAILED.'
056547   200   123   110                          .ASCIZ  <CRLF>'SHOULD HAVE GONE FROM STATE 420 TO 131.'
056620                                  EM155:
056620   101   116   040                          .ASCII  'AN UNDERFLOW ERROR OCCURRED ON ADD'<CRLF>
056663   103   101   125                          .ASCII  'CAUSING A TRAP TO 244.'
056711   200   050   102                          .ASCII  <CRLF>'(BUT EZBT Y9 Y8) FORK IN STATE 420 OF OVER\UNDER FAILED.'
057002   200   123   110                          .ASCIZ  <CRLF>'SHOULD HAVE GONE FROM STATE 420 TO 131.'
057053                                  EM156:
057053   101   104   104                          .ASCII  /ADDD (R0),AC0 FAILED IN THE ROUND\TRUNK FLOWS./
057131   200   124   110                          .ASCII  <CRLF>'THE (BUT FD) FORK FAILED. WENT'
057170   106   122   117                          .ASCII  \FROM STATE 665 TO 113.\<CRLF>
057217   111   116   123                          .ASCIZ  \INSTEAD OF FROM 665 TO 313.\<CRLF>\WITH FT SET.\
057270                                  EM157:
057270   101   104   104                          .ASCII  /ADDD (R0),AC0 FAILED IN THE ROUND\TRUNK FLOWS./
057346   200   124   110                          .ASCII  <CRLF>'THE (BUT FD) FORK FAILED. WENT'
057405   106   122   117                          .ASCII  \FROM STATE 665 TO 313.\<CRLF>
057434   111   116   123                          .ASCIZ  \INSTEAD OF FROM 665 TO 113.\<CRLF>\WITH FT CLEAR.\
057507                                  EM160:
057507   101   104   104                          .ASCII  /ADDD (R0),AC0 FAILED IN THE ROUND\TRUNK FLOWS./<CRLF>
057566   124   110   105                          .ASCII  'THE FLOATING CONSTANT WAS USED INSTEAD OF THE DOUBLE CONSTANT'<CRLF>
057664   111   116   040                          .ASCIZ  'IN THE ROUND ALGORITHM.'
057714                                  EM161:
057714   101   104   104                          .ASCII  /ADDF (R0),AC0 FAILED IN THE ROUND\TRUNK FLOWS./<CRLF>
057773   124   110   105                          .ASCII  'THE DOUBLE CONSTANT WAS USED INSTEAD OF THE FLOATING CONSTANT'<CRLF>
060071   111   116   040                          .ASCIZ  'IN THE ROUND ALGORITHM.'
060121                                  EM162:
060121   101   104   104                          .ASCIZ  /ADDD (R0),AC0 PRODUCED A BAD RESULT./
060166                                  EM163:
060166   101   104   104                          .ASCIZ  /ADDF (R0),AC0 PRODUCED A BAD RESULT./
060233                                  EM164:
060233   124   110   105                          .ASCIZ  \THE FPS WAS BAD AFTER ADDD (R0),AC0.\
060300                                  EM165:
060300   124   110   105                          .ASCIZ  \THE FPS WAS BAD AFTER ADDF (R0),AC0.\
060345                                  EM166:
060345   101   104   104                          .ASCII  /ADDD (R0),AC0 PRODUCED A BAD RESULT,/<CRLF>
060412   120   122   117                          .ASCIZ  'PROBABLE ERROR IN THE ALIGN FLOWS.'
060455                                  EM167:
060455   101   104   104                          .ASCII  /ADDD (R0),AC0 FAILED IN THE ALIGN FLOWS./<CRLF>
060526   106   114   117                          .ASCII  \FLOW DID NOT FOLLOW THE PATH: STATE 476, TO 111, TO 014.\
060616   200   101   040                          .ASCII  <CRLF>\A BAD CONSTANT (NOT 57 DEC) \
060653   127   101   123                          .ASCIZ  'WAS USED IN THE ALIGN ALGORITHM.'
060714                                  EM170:
060714   101   104   104                          .ASCII  /ADDF (R0),AC0 PRODUCED A BAD RESULT,/<CRLF>
060761   120   122   117                          .ASCIZ  'PROBABLE ERROR IN THE ALIGN FLOWS.'
061024                                  EM171:
061024   101   104   104                          .ASCII  /ADDF (R0),AC0 FAILED IN THE ALIGN FLOWS./<CRLF>
061075   106   114   117                          .ASCII  \FLOW DID NOT FOLLOW THE PATH: STATE 476, TO 111, TO 014.\
```

ERROR MESSAGES

```
        061165  200  101  040            .ASCII  <CRLF>\A BAD CONSTANT (NOT 25 DEC) \
        061222  127  101  123            .ASCIZ  'WAS USED IN THE ALIGN ALGORITHM.'
 5978   061263                    EM172:
        061263  101  104  104            .ASCII  /ADDD (R0),AC0 FAILED IN THE ALIGN FLOWS./<CRLF>
        061334  106  114  117            .ASCII  \FLOW DID NOT FOLLOW THE PATH: STATE 476, TO 011, TO 015.\
        061424  200  101  040            .ASCII  <CRLF>\A BAD CONSTANT (NOT 57 DEC) \
        061461  127  101  123            .ASCIZ  'WAS USED IN THE ALIGN ALGORITHM.'
 5979   061522                    EM173:
        061522  101  104  104            .ASCII  /ADDD (R0),AC0 FAILED IN THE ALIGN FLOWS./<CRLF>
        061573  106  114  117            .ASCII  \FLOW DID NOT FOLLOW THE PATH: STATE 476, TO 011, TO 215.\
        061663  200  101  040            .ASCII  <CRLF>\A BAD CONSTANT (NOT 57 DEC) \
        061720  127  101  123            .ASCIZ  'WAS USED IN THE ALIGN ALGORITHM.'
 5980   061761                    EM174:
        061761  101  104  104            .ASCII  /ADDF (R0),AC0 FAILED IN THE ALIGN FLOWS./<CRLF>
        062032  106  114  117            .ASCII  \FLOW DID NOT FOLLOW THE PATH: STATE 476, TO 011, TO 015.\
        062122  200  101  040            .ASCII  <CRLF>\A BAD CONSTANT (NOT 25 DEC) \
        062157  127  101  123            .ASCIZ  'WAS USED IN THE ALIGN ALGORITHM.'
 5981   062220                    EM175:
        062220  101  104  104            .ASCII  /ADDF (R0),AC0 FAILED IN THE ALIGN FLOWS./<CRLF>
        062271  106  114  117            .ASCII  \FLOW DID NOT FOLLOW THE PATH: STATE 476, TO 011, TO 215.\
        062361  200  101  040            .ASCII  <CRLF>\A BAD CONSTANT (NOT 25 DEC) \
        062416  127  101  123            .ASCIZ  'WAS USED IN THE ALIGN ALGORITHM.'
 5982   062457                    EM176:
        062457  101  104  104            .ASCII  'ADDD (R0),AC0 FAILED IN THE ADD-SUB FLOWS.'<CRLF>
        062532  104  111  104            .ASCIZ  \DID NOT TAKE THE PATH: STATE 216, TO 442, TO 500.\
 5983   062614                    EM177:
        062614  101  104  104            .ASCII  'ADDD (R0),AC0 FAILED IN THE ADD-SUB FLOWS.'<CRLF>
        062667  104  111  104            .ASCIZ  \DID NOT TAKE THE PATH: STATE 216, TO 042, TO 121.\
 5984   062751                    EM200:
        062751  101  104  104            .ASCII  'ADDD (R0),AC0 FAILED IN THE ADD-SUB FLOWS.'<CRLF>
        063024  104  111  104            .ASCIZ  \DID NOT TAKE THE PATH: STATE 216, TO 440, TO 121.\
 5985   063106                    EM201:
        063106  101  104  104            .ASCII  'ADDD (R0),AC0 FAILED IN THE ADD-SUB FLOWS.'<CRLF>
        063161  104  111  104            .ASCIZ  \DID NOT TAKE THE PATH: STATE 216, TO 440, TO 101.\
 5986   063243                    EM202:
        063243  101  104  104            .ASCII  'ADDD (R0),AC0 FAILED IN THE ADD-SUB FLOWS.'<CRLF>
        063316  104  111  104            .ASCIZ  \DID NOT TAKE THE PATH: STATE 216, TO 042, TO 101.\
 5987   063400                    EM203:
        063400  101  104  104            .ASCII  'ADDD (R0),AC0 FAILED IN THE ADD-SUB FLOWS.'<CRLF>
        063453  104  111  104            .ASCIZ  \DID NOT TAKE THE PATH: STATE 216, TO 440, TO 141.\
 5988   063535                    EM204:
        063535  101  104  104            .ASCII  'ADDD (R0),AC0 FAILED IN THE ADD-SUB FLOWS.'<CRLF>
        063610  104  111  104            .ASCIZ  \DID NOT TAKE THE PATH: STATE 216, TO 042, TO 141.\
 5989   063672                    EM205:
        063672  124  110  105            .ASCIZ  \THE FPS WAS BAD AFTER SUBD (R0),AC0.\
 5990   063737                    EM206:
        063737  123  125  102            .ASCIZ  /SUBD (R0),AC0 PRODUCED A BAD RESULT./
 5991   064004  123  125  102    EM207:  .ASCII  'SUBD (R0),AC0 PRODUCED A BAD RESULT.'
 5992   064050  200  124  110            .ASCIZ  <CRLF>'THE XOR OF THE SIGN BIT FAILED IN STATE 024.'
 5993   064126  101  104  104    EM210:  .ASCIZ  'ADDD (R0),AC0 FAILED IN THE NORMALIZE FLOWS.'
 5994          044166                    EM211=EM5
 5995          044222                    EM212=EM6
 5996          044254                    EM213=EM7
```

DATA HEADERS

```
                                                        .SBTTL  DATA HEADERS
064203      040     040     124     DH1:    .ASCII  '  TEST.'<TAB>'PC OF CALL.'<TAB>'PC OF ERROR.'
064243      011     127     122             .ASCIZ  <TAB>'WROTE.'<TAB>'READ.'<TAB>'EXPECTED.'
064273      040     040     124     DH2:    .ASCII  '  TEST.'<TAB>'PC OF CALL.'<TAB>'PC OF ERROR.'
064333      101     116     104             .ASCIZ  'AND BAD DATA.'<TAB>'OR BAD DATA.'
064366      040     040     124     DH3:    .ASCII  '  TEST.'<TAB>'PC OF CALL.'<TAB>'PC OF ERROR.'
064426      011     122     105             .ASCIZ  <TAB>'READ PSW.'<TAB>'EXPECTED PSW.'
064457      040     040     124     DH4:    .ASCII  '  TEST.'<TAB>'PC OF CALL.'<TAB>'PC OF ERROR.'
064517      011     127     122             .ASCIZ  <TAB>'WROTE FPS.'<TAB>'FPS AFTER CFCC.'
064553      040     040     124     DH5:    .ASCIZ  '  TEST.'<TAB>'PC OF CALL.'<TAB>'PC OF TRAP.'
            064553                  DH6=DH5
            064553                  DH7=DH5
            064553                  DH10=DH5
            064553                  DH11=DH5
            000000                  DH12=0
            000000                  DH13=0
            064553                  DH14=DH5
            064553                  DH15=DH5
064613      040     040     124     DH16:   .ASCII  '  TEST.'<TAB>'PC OF CALL.'<TAB>'PC OF ERROR.'
064653      011     117     120             .ASCIZ  <TAB>'OP CODE.  FPS.'
064673      040     040     124     DH17:   .ASCII  '  TEST.'<TAB>'PC OF CALL.'<TAB>'PC OF ERROR.'
064733      011     107     117             .ASCIZ  <TAB>'GOT FPS.'<TAB>'EXPECTED FPS.'
064763      040     040     124     DH20:   .ASCII  '  TEST.'<TAB>'PC OF CALL.'<TAB>'PC OF TRAP.'
065022      011     120     103             .ASCIZ  <TAB>'PC OF STST.'<TAB>'READ FEC.'
            064553                  DH21=DH5
065051      106     101     111     DH22:   .ASCIZ  'FAILED TO CORRECTLY SET FPS.'
065106      106     101     111     DH23:   .ASCII  'FAILED TO CORRECTLY SET FEC TO 000002.'<CRLF>
065155      040     040     124             .ASCII  '  TEST.'<TAB>'PC OF CALL.'<TAB>'PC OF ERROR.'
065215      011     120     103             .ASCIZ  <TAB>'PC OF STST.'<TAB>'READ FEC.'
065244      124     122     101     DH24:   .ASCII  'TRAPPED TO 244. FLOW WENT FROM STATE 554 TO STATE 430.'
065332      200     111     116             .ASCIZ  <CRLF>'INSTEAD OF FROM STATE 554 TO STATE 432.'
065403      040     040     124     DH25:   .ASCIZ  '  TEST.'<IAB>'PC OF CALL.'<TAB>'PC OF ERROR.'<TAB>
065445      040     040     124     DH26:   .ASCII  '  TEST.'<TAB>'PC OF CALL.'<TAB>'PC OF ERROR.'
065505      011     107     117             .ASCIZ  <TAB>'GOT R0.'<TAB>'EXPECTED R0.'
            065445                  DH27=DH26
            000000                  DH30=0
            065445                  DH31=DH26
            065445                  DH32=DH26
065533      040     040     124     DH33:   .ASCII  '  TEST.'<TAB>'PC OF CALL.'<TAB>'PC OF ERROR.'
065573      011     122     060             .ASCIZ  <TAB>'R0 (TARGET LOCATIONS FOR OUTPUT).'
            065533                  DH34=DH33
            065533                  DH35=DH33
            065533                  DH36=DH33
            065533                  DH37=DH33
            065533                  DH40=DH33
            000000                  DH41=0
065636      040     040     124     DH42:   .ASCII  '  TEST.'<TAB>'PC OF CALL.'<TAB>'PC OF TRAP.'
065675      011     122     060             .ASCIZ  <TAB>'R0 (TARGET LOCATIONS FOR OUTPUT).'
            065636                  DH43=DH42
            000000                  DH44=0
065740      105     122     122     DH45:   .ASCIZ  'ERROR SUMMARY.'
06575?      040     040     124     DH46:   .ASCIZ  '  TEST.'<TAB>'CALL AT PC.'
            065740                  DH47=DH45
066003      040     040     124     DH50:   .ASCII  '  TEST.'<TAB>'PC OF CALL.'<TAB>'PC OF ERROR.'
066043      011     127     111             .ASCIZ  <TAB>'WITH FD.'
            066003                  DH51=DH50
            064553                  DH52=DH5
```

```
                065445                  DH53=DH26
066055  040     040     124     DH54:   .ASCIZ   '  TEST.'<TAB>'PC OF CALL.'<TAB>'PC OF ERROR.'
                064553                  DH55=DH5
                065445                  DH56=DH26
                066055                  DH57=DH54
                065445                  DH60=DH26
                066055                  DH61=DH54
                066055                  DH62=DH54
                066055                  DH63=DH54
066116  122     105     123     DH65:   .ASCII   'RESULTING IN AN ODD ADDRESS TRAP TO 4.'
066164  200                             .ASCII   <CRLF>
066165  040     040     124     DH64:   .ASCII   '  TEST.'<TAB>'PC OF CALL.'<TAB>'PC OF ERROR.'
066225  011     107     117             .ASCIZ   <TAB>'GOT PC.'<TAB>'EXPECTED PC.'
                066055                  DH66=DH54
                064553                  DH67=DH5
                064553                  DH70=DH5
                065403                  DH71=DH25
                065445                  DH72=DH26
                066055                  DH73=DH54
                064553                  DH74=DH5
                064553                  DH75=DH5
                065403                  DH76=DH25
                065445                  DH77=DH26
                066055                  DH100=DH54
                064553                  DH101=DH5
                065445                  DH102=DH26
                065403                  DH103=DH25
                066055                  DH104=DH54
                064553                  DH105=DH5
                065445                  DH106=DH26
                065403                  DH107=DH25
                066055                  DH110=DH54
                065403                  DH111=DH25
066253  124     110     105     DH112:  .ASCII   'THE (BUT FSRC) FORK FAILED.'<CRLF>
066307  103     117     116             .ASCII   'CONTROL WENT FROM STATE 762 TO STATE 627.'
066360  200     111     116             .ASCII   <CRLF>'INSTEAD OF FROM STATE 762 TO STATE 637.'<CRLF>
066431  040     040     124             .ASCIZ   '  TEST.'<TAB>'PC OF CALL.'<TAB>'PC OF ERROR.'
                065403                  DH113=DH25
                066253                  DH114=DH112
066472  124     110     105     DH115:  .ASCII   'THE (BUT FSRC) FORK FAILED RESULTING IN AN ODD ADDRESS TRAP TO 4.'
066573  200     103     117             .ASCII   <CRLF>'CONTROL WENT FROM STATE 762 TO STATE 627.'<CRLF>
066646  111     116     123             .ASCII   'INSTEAD OF FROM STATE 762 TO STATE 627.'<CRLF>
066716  040     040     124             .ASCIZ   '  TEST.'<TAB>'PC OF CALL.'<TAB>'PC OF TRAP.'
                066472                  DH116=DH115
                064673                  DH117=DH17
066756  040     040     124     DH120:  .ASCII   '  TEST.'<TAB>'PC OF CALL.'<TAB>'PC OF ERROR.'
067016  011     107     117             .ASCIZ   <TAB>'GOT FEC.'<TAB>'EXPFCTED FEC.'
                064553                  DH121=DH5
                066003                  DH122=DH50
                066003                  DH123=DH50
                064673                  DH124=DH17
                064673                  DH125=DH17
                064553                  DH126=DH5
                066055                  DH127=DH54
                066756                  DH130=DH120
                066055                  DH131=DH54
                066055                  DH132=DH54
```

```
 6111 067046     106     101     111  DH133:  .ASCII  'FAILED TO PRODUCE THE CORRECT RESULTS.'<CRLF>
 6112 067115     040     040     124          .ASCIZ  '  TEST.'<TAB>'PC OF CALL.'<TAB>'PC OF ERROR.'
 6113         067046                  DH134=DH133
 6114         067046                  DH135=DH133
 6115         067046                  DH136=DH133
 6116 067156     120     122     117  DH137:  .ASCII  'PRODUCED THE CORRECT RESULT BUT FAILED TO SET THE FPS CORRECTLY.'
 6117 067256     040     040     124          .ASCII  '  TEST.'<TAB>'PC OF CALL.'<TAB>'PC OF ERROR.'
 6118 067316     011     107     117          .ASCIZ  <TAB>'GOT FPS.'<TAB>'EXPECTED FPS.'
 6119         067156                  DH140=DH137
 6120         067156                  DH141=DH137
 6121         067156                  DH142=DH137
 6122         067046                  DH143=DH133
 6123         067046                  DH144=DH133
 6124         067156                  DH145=DH137
 6125         067156                  DH146=DH137
 6126         066055                  DH147=DH54
 6127 067346     130     117     122  DH150:  .ASCII  'XOR OF SIGN BIT FAILED.'<CRLF>
 6128 067376     040     040     124          .ASCIZ  '  TEST.'<TAB>'PC OF CALL.'<TAB>'PC OF ERROR.'
 6129         067156                  DH151=DH137
 6130         067046                  DH152=DH133
 6131         067156                  DH153=DH137
 6132 067437     040     040     124  DH154:  .ASCIZ  '  TEST.'<TAB>'PC OF CALL.'<TAB>'PC OF TRAP.'
 6133         067437                  DH155=DH154
 6134         066055                  DH156=DH54
 6135         066055                  DH157=DH54
 6136         066055                  DH160=DH54
 6137         066055                  DH161=DH54
 6138         066055                  DH162=DH54
 6139         066055                  DH163=DH54
 6140         064673                  DH164=DH17
 6141         064673                  DH165=DH17
 6142         066055                  DH166=DH54
 6143         066055                  DH167=DH54
 6144         066055                  DH170=DH54
 6145         066055                  DH171=DH54
 6146         066055                  DH172=DH54
 6147         066055                  DH173=DH54
 6148         066055                  DH174=DH54
 6149         066055                  DH175=DH54
 6150         066055                  DH176=DH54
 6151         066055                  DH177=DH54
 6152         066055                  DH200=DH54
 6153         066055                  DH201=DH54
 6154         066055                  DH202=DH54
 6155         066055                  DH203=DH54
 6156         066055                  DH204=DH54
 6157         064673                  DH205=DH17
 6158         066055                  DH206=DH54
 6159         066055                  DH207=DH54
 6160         066055                  DH210=DH54
 6161 067477     040     040     124  DH211:  .ASCIZ  '  TEST.'<TAB>'PC OF CALL.'<TAB>'PC OF TRAP.'<TAB>'FEC.'
 6162         064553                  DH212=DH5
 6163         064553                  DH213=DH5
```

DATA FORMATS

```
                                                    .SBTTL  DATA FORMATS
067544      004     000     005     DF1:    .BYTE   4,0,5,0,5,0,0,0
067554      004     000     005     DF2:    .BYTE   4,0,5,4,5,0,5,0
067564      004     000     005     DF3:    .BYTE   4,0,5,0,5,0,5,0
        067564                      DF4=DF3
067574      004     000     005     DF5:    .BYTE   4,0,5,0,5,0,5,11,5,0,5,0
        067574                      DF6=DF5
        067574                      DF7=DF5
        067574                      DF10=DF5
        067574                      DF11=DF5
067610      005     011     005     DF12:   .BYTE   5,11,5,5,5,4,5,4,5,5,4,5,4,5,11,5,11,5,5,4,0,5,0,5,0,0
067642      005     011     005     DF13:   .BYTE   5,11,5,5,5,4,0,5,0,5,0,0
        067574                      DF14=DF6
        067574                      DF15=DF6
067656      004     000     005     DF16:   .BYTE   4,0,5,0,5,0,0
        067564                      DF17=DF3
067665      004     000     005     DF20:   .BYTE   4,0,5,0,5,0,5,0
067675      004     000     005     DF21:   .BYTE   4,0,5,0
067701      005     005     004     DF22:   .BYTE   5,5,4,0,5,0,5,0,5,0
067713      004     000     005     DF23:   .BYTE   4,0,5,0,5,0,5,0
067723      005     004     000     DF24:   .BYTE   5,4,0,5,0,5,0
067732      004     000     005     DF25:   .BYTE   4,0,5,0,5,5,5,0,5,0,5,0,5,5,5,0,5,0,5,0
067756      004     000     005     DF26:   .BYTE   4,0,5,0,5,0,0,5,5,5,5,4,5,4,5,5,5,5,4,5,4
070003      004     000     005     DF27:   .BYTE   4,0,5,0,5,0,0,5,5,5,5,4,5,4,5,5
070023      005     011     005     DF30:   .BYTE   5,11,5,5,5,4,0,5,0,5,0,0
        067756                      DF31=DF26
        070003                      DF32=DF27
070037      004     000     005     DF33:   .BYTE   4,0,5,0,5,0,5,5,5,0,5,0,5,12,5,5,5,0,5,0,5,12
070065      004     000     005     DF34:   .BYTE   4,0,5,0,5,0,5,5,5,5,4,5,4,5,5,5,5,4,5,4
        070065                      DF35=DF34
        070065                      DF36=DF34
        070065                      DF37=DF34
070111      004     000     005     DF40:   .BYTE   4,0,5,0,5,0,5,5,5,0,5,0,5,3,5,5,5,0,5,0,5,3
070137      011     005     005     DF41:   .BYTE   11,5,5,5,4,0,5,0,5,0,5,0
070153      004     000     005     DF42:   .BYTE   4,0,5,0,5,0,5,5,5,5,4,5,4,11,4,5,5,5,5,4,5,4
        070153                      DF43=DF42
070201      005     011     005     DF44:   .BYTE   5,11,5,5,5,4,0,5,0,5,5,5,5,3,5,5,5,5,3
070224      005     011     005     DF45:   .BYTE   5,11,5,5,5,4,0,5,0,5,4,5,5,5,5,3,5,5,5,5,3
070251      004     000     005     DF46:   .BYTE   4,0,5,5,5,3,5,3,5,3,5,3,5,3,5,5,5,3,5,3,5,3,5,3,5,3
070303      004     000     005     DF47:   .BYTE   4,0,5,0,5,5,5,4,5,4,5,5
070320      004     000     005     DF50:   .BYTE   4,0,5,0,5,11,5,5,5,5,4,5,4,5,5,5,5,4,5,4
070344      004     000     005     DF51:   .BYTE   4,0,5,0,5,11,5,5,5,3,5,5,5,3
        070303                      DF52=DF47
070362      004     000     005     DF53:   .BYTE   4,0,5,0,5,0,0
070371      004     000     005     DF54:   .BYTE   4,0,5,0,5,5,5,5,3,5,5,5,5,3
        070303                      DF55=DF47
        070362                      DF56=DF53
        070371                      DF57=DF54
        070362                      DF60=DF53
        070371                      DF61=DF54
        070371                      DF62=DF54
        070371                      DF63=DF54
070407      004     000     005     DF64:   .BYTE   4,0,5,0,5,0,0
        070407                      DF65=DF64
        070371                      DF66=DF54
        067675                      DF67=DF21
070416      004     000     005     DF70:   .BYTE   4,0,5,0,5,5,5,5,4,5,4,5,5,5,5,4,5,4
```

DATA FORMATS

```
6221            070416                          DF71=DF70
6222 070440       004       000       005       DF72:    .BYTE   4,0,5,0,5,0,0
6223            070371                          DF73=DF54
6224            067675                          DF74=DF21
6225            070416                          DF75=DF70
6226            070416                          DF76=DF70
6227            070440                          DF77=DF72
6228            070371                          DF100=DF54
6229            070416                          DF101=DF70
6230            070440                          DF102=DF72
6231            070416                          DF103=DF70
6232            070371                          DF104=DF54
6233            070416                          DF105=DF70
6234            070440                          DF106=DF72
6235            070416                          DF107=DF70
6236            070371                          DF110=DF54
6237 070447       004       000       005       DF111:   .BYTE   4,0,5,0
6238            070447                          DF112=DF111
6239            070447                          DF113=DF111
6240            070447                          DF114=DF111
6241            070447                          DF115=DF111
6242            070447                          DF116=DF111
6243            067564                          DF117=DF3
6244            067564                          DF120=DF3
6245            070303                          DF121=DF47
6246            070320                          DF122=DF50
6247            070344                          DF123=DF51
6248 070453       004       000       005       DF124:   .BYTE   4,0,5,0,5,0,0,5,5,5,5,4,5,4,5,5,5,5,4,5,4,5,5,5,3
6249            070453                          DF125=DF124
6250            070447                          DF126=DF111
6251            070447                          DF127=DF111
6252            067564                          DF130=DF3
6253 070504       004       000       005       DF131:   .BYTE   4,0,5,0,5,5,5,3,5,5,5,3,5,5,5,3
6254            070504                          DF132=DF131
6255 070524       004       000       005       DF133:   .BYTE   4,0,5,0,5,5,5,3,5,5,5,3,5,5,5,3,5,5,5,3
6256            070524                          DF134=DF133
6257            070524                          DF135=DF133
6258            070524                          DF136=DF133
6259 070550       004       000       005       DF137:   .BYTE   4,0,5,0,5,0,5,0
6260            070550                          DF140=DF137
6261            070550                          DF141=DF137
6262            070550                          DF142=DF137
6263            070524                          DF143=DF133
6264            070524                          DF144=DF133
6265            070550                          DF145=DF137
6266            070550                          DF146=DF137
6267            070524                          DF147=DF133
6268            070524                          DF150=DF133
6269            070550                          DF151=DF137
6270            070524                          DF152=DF133
6271            070550                          DF153=DF137
6272 070560       004       000       005       DF154:   .BYTE   4,0,5,0
6273            070560                          DF155=DF154
6274            070524                          DF156=DF133
6275            070524                          DF157=DF133
6276            070524                          DF160=DF133
6277 070564       004       000       005       DF161:   .BYTE   4,0,5,0,5,5,5,2,5,5,5,2,5,5,5,2,5,5,5,2
```

```
 6278            070524                  DF162=DF133
 6279            070564                  DF163=DF161
 6280            067564                  DF164=DF3
 6281            067564                  DF165=DF3
 6282            070524                  DF166=DF133
 6283            070524                  DF167=DF133
 6284            070564                  DF170=DF161
 6285            070564                  DF171=DF161
 6286            070524                  DF172=DF133
 6287            070524                  DF173=DF133
 6288            070564                  DF174=DF161
 6289            070564                  DF175=DF161
 6290            070524                  DF176=DF133
 6291            070524                  DF177=DF133
 6292            070524                  DF200=DF133
 6293            070524                  DF201=DF133
 6294            070524                  DF202=DF133
 6295            070524                  DF203=DF133
 6296            070524                  DF204=DF133
 6297            067564                  DF205=DF3
 6298            070524                  DF206=DF133
 6299            070524                  DF207=DF133
 6300            070524                  DF210=DF133
 6301 070610       004       000     005  DF211:  .BYTE   4,0,5,0,5,0
 6302            070610                  DF212=DF211
 6303            070610                  DF213=DF211
 6304                                    .EVEN
```

DATA TABLES

```
                                                        .SBTTL  DATA TABLES
070616  001232  001234  041346  DT1:    .WORD   $TMP0,$TMP1,$TAB,$TMP2,$TAB,$TMP3
070632  001242  001244  000000          .WORD   $TMP4,$TMP5,0
070640  001232  001234  041346  DT2:    .WORD   $TMP0,$TMP1,$TAB,AERFLG,$TAB,$TMP2,$TAB,$TMP3,0
070662  001232  001234  041346  DT3:    .WORD   $TMP0,$TMP1,$TAB,$TMP2,$TAB,$TMP3
070676  041346  001242  000000          .WORD   $TAB,$TMP4,0
        070662                  DT4=DT3
070704  001232  001234  041346  DT5:    .WORD   $TMP0,$TMP1,$TAB,$TMP2,LFIEX1,$TMP21,LFIEX2
070722  001272  041465  001240          .WORD   $TMP20,FPSMS,$TMP3,FECMS,$TMP4,0
070736  001232  001234  041346  DT6:    .WORD   $TMP0,$TMP1,$TAB,$TMP2,LFIEX1,$TMP21,LFIEX2,$TMP20,0
        070736                  DT7=DT6
        070736                  DT10=DT6
        070736                  DT11=DT6
070760  041542  001252  041547  DT12:   .WORD   $THE,$TMP10,NOOP1,NOOP15,NOOP2,$TMP5
071016  041710  001246  041722          .WORD   NOOP3,$TMP6,NOOP4,NOOP2,$TMP5,NOOP3,$TMP7,NOOP5,$TMP11
071036  041765  001252  042005          .WORD   NOOP6,$TMP10,NOOP7,NOOP10,$TMP0,$TMP1,$TAB,$TMP2
071046  041346  001240  001242          .WORD   $TAB,$TMP3,$TMP4,0
071066  041542  001252  041547  DT13:   .WORD   $THE,$TMP10,NOOP1,NOOP11,NOOP10,$TMP0,$TMP1,$TAB
        001236  041346  001240          .WORD   $TMP2,$TAB,$TMP3,$TMP4,0
        070736                  DT14=DT6
        070736                  DT15=DT6
071100  001232  001234  041346  DT16:   .WORD   $TMP0,$TMP1,$TAB,$TMP2,$TAB,$TMP5,$TMP3,0
        070662                  DT17=DT3
071120  001232  001234  041346  DT20:   .WORD   $TMP0,$TMP1,$TAB,$TMP2,$TAB,$TMP3
071134  041346  001242  000000          .WORD   $TAB,$TMP4,0
071142  001232  001234  041346  DT21:   .WORD   $TMP0,$TMP1,$TAB,$TMP2,0
071154  064366  001313          DT22:   .WORD   DH3,$CRLF
071160  001232  001234  041346          .WORD   $TMP0,$TMP1,$TAB,$TMP2,$TAB,$TMP3
071174  041346  001242  000000          .WORD   $TAB,$TMP4,0
071202  001232  001234  041346  DT23:   .WORD   $TMP0,$TMP1,$TAB,$TMP2,$TAB,$TMP3
071216  041346  001242  000000          .WORD   $TAB,$TMP4,0
071224  042441                  DT24:   .WORD   ILLMS
071226  001232  001234  041346          .WORD   $TMP0,$TMP1,$TAB,$TMP2,$TAB,$TMP3
071242  041346  001242  000000          .WORD   $TAB,$TMP4,0
071250  001232  001234  041346  DT25:   .WORD   $TMP0,$TMP1,$TAB,$CRLF,MS1,MS3,$TMP3,MS4,$TMP4,$CRLF
071274  001242  001313  042521          .WORD   $TMP4,$CRLF,MS2,MS3,$TMP5,MS4,$TMP6,$CRLF,$TMP5,0
071320  001232  001234  041346  DT26:   .WORD   $TMP0,$TMP1,$TAB,$TMP2,$TAB,$TMP3,$TMP4,$CRLF
071340  042631  001313  042647          .WORD   MS6,$CRLF,MS7,$TMP5,MS10,$TMP6,$CRLF
071356  042676  001313  042647          .WORD   MS11,$CRLF,MS7,$TMP5,MS10,$TMP7,0
071374  001232  001234  041346  DT27:   .WORD   $TMP0,$TMP1,$TAB,$TMP2,$TAB,$TMP3
071410  001242  001313  042723          .WORD   $TMP4,$CRLF,MS12,$CRLF,MS7,$TMP5,MS10,$TMP7,$CRLF,MS13,0
071436  043064  001272  001313  DT30:   .WORD   MS15,$TMP20,$CRLF,MS14,$CRLF
071450  001232  001234  041346          .WORD   $TMP0,$TMP1,$TAB,$TMP2,$TAB,$TMP3
071464  001242  000000                  .WORD   $TMP4,0
        071320                  DT31=DT26
        071374                  DT32=DT27
071470  001232  001234  041346  DT33:   .WORD   $TMP0,$TMP1,$TAB,$TMP2,$TAB,$TMP3
071504  001313  042507  042526          .WORD   $CRLF,MS1,MS3,$TMP4,MS4,$TMP5,$CRLF,$TMP6,$CRLF
071526  042521  042526  001242          .WORD   MS2,MS3,$TMP4,MS4,$TMP5,$CRLF,$TMP4,0
071546  001232  001234  041346  DT34:   .WORD   $TMP0,$TMP1,$TAB,$TMP2,$TAB,$TMP3
071562  001313  042631  001313          .WORD   $CRLF,MS6,$CRLF,MS7,$TMP5,MS10,$TMP6,$CRLF
071602  042676  001313  042647          .WORD   MS11,$CRLF,MS7,$TMP5,MS10,$TMP7,0
        071546                  DT35=DT34
        071546                  DT36=DT34
        071546                  DT37=DT34
        071470                  DT40=DT33
071620  001272  001313  064366  DT41:   .WORD   $TMP20,$CRLF,DH3,$CRLF
```

```
6362 071630  001232  001234  041346          .WORD   $TMP0,$TMP1,$TAB,$TMP2,$TAB,$TMP3
6363 071644  041346  001242  000000          .WORD   $TAB,$TMP4,0
6364 071652  001232  001234  041346  DT42:   .WORD   $TMP0,$TMP1,$TAB,$TMP2,$TAB,$TMP3
6365 071666  001313  042631  001313          .WORD   $CRLF,MS6,$CRLF,MS7,$TMP5,MS10,$TMP6,$TMP15,$TMP10
6366 071710  001313  042676  001313          .WORD   $CRLF,MS11,$CRLF,MS7,$TMP5,MS10,$TMP7,0
6367         071652                  DT43=DT42
6368 071730  043116  001244  001313  DT44:   .WORD   MS17,$TMP5,$CRLF,MS20,$CRLF,$TMP0,$TMP1,$TAB,$TMP2
6369 071752  001313  042507  043247          .WORD   $CRLF,MS1,MS21,$CRLF,$TMP3,$CRLF,MS2,MS21,$CRLF,$TMP4,0
6370 072000  043116  001244  001313  DT45:   .WORD   MS17,$TMP5,$CRLF,MS24,$CRLF,$TMP0,$TMP1,$TAB,$TMP2,$TAB
6371 072024  001246  001313  043306          .WORD   $TMP6,$CRLF,MS22,MS21,$CRLF,$TMP3,$CRLF
6372 072042  043336  043247  001313          .WORD   MS23,MS21,$CRLF,$TMP4,0
6373 072054  001232  001234  001313  DT46:   .WORD   $TMP0,$TMP1,$CRLF,MS25,MS30,$TMP2,MS31,$TMP3
6374 072074  043544  001242  043553          .WORD   MS32,$TMP4,MS33,$TMP5,MS34,$TMP6,$CRLF,MS26
6375 072114  043526  001250  043535          .WORD   MS30,$TMP7,MS31,$TMP10
6376 072124  043544  001254  043553          .WORD   MS32,$TMP11,MS33,$TMP12,MS34,$TMP13,0
6377 072142  001232  001234  041346  DT47:   .WORD   $TMP0,$TMP1,$TAB,$TMP2,$CRLF,MS12,MS7,$TMP3,MS10
6378 072164  001242  001313  042761          .WORD   $TMP4,$CRLF,MS13,0
6379         071546                  DT50=DT34
6380 072174  001232  001234  041346  DT51:   .WORD   $TMP0,$TMP1,$TAB,$TMP2,$TAB,$TMP3
6381 072210  001313  043460  043517          .WORD   $CRLF,MS25,MS27,$TMP4,$CRLF,MS26,MS27,$TMP5,0
6382         072142                  DT52=DT47
6383 072232  001232  001234  041346  DT53:   .WORD   $TMP0,$TMP1,$TAB,$TMP2,$TAB,$TMP3
6384 072246  001242  000000          .WORD   $TMP4,0
6385 072252  001232  001234  041346  DT54:   .WORD   $TMP0,$TMP1,$TAB,$TMP2,$CRLF,MS1,MS21,$CRLF,$TMP3
6386 072274  001313  042521  043247          .WORD   $CRLF,MS2,MS21,$CRLF,$TMP3,0
6387         072142                  DT55=DT47
6388         072232                  DT56=DT53
6389         072252                  DT57=DT54
6390         072232                  DT60=DT53
6391         072252                  DT61=DT54
6392         072252                  DT62=DT54
6393         072252                  DT63=DT54
6394 072310  001232  001234  041346  DT64:   .WORD   $TMP0,$TMP1,$TAB,$TMP2,$TAB,$TMP3
6395 072324  001242  000000          .WORD   $TMP4,0
6396         072310                  DT65=DT64
6397         072252                  DT66=DT54
6398         071142                  DT67=DT21
6399 072330  001232  001234  041346  DT70:   .WORD   $TMP0,$TMP1,$TAB,$TMP2,$CRLF,MS6,$CRLF,MS7,$TMP5
6400 072352  042663  001246  001313          .WORD   MS10,$TMP6,$CRLF,MS11,$CRLF,MS7,$TMP5,MS10,$TMP7,0
6401         072330                  DT71=DT70
6402 072376  001232  001234  041346  DT72:   .WORD   $TMP0,$TMP1,$TAB,$TMP2,$TAB,$TMP3,$TMP4,0
6403         072252                  DT73=DT54
6404         071142                  DT74=DT21
6405         072330                  DT75=DT70
6406         072330                  DT76=DT70
6407         072376                  DT77=DT72
6408         072252                  DT100=DT54
6409         072330                  DT101=DT70
6410         072330                  DT102=DT71
6411         072330                  DT103=DT70
6412         072252                  DT104=DT54
6413         072330                  DT105=DT70
6414         072376                  DT106=DT72
6415         072330                  DT107=DT70
6416         072252                  DT110=DT54
6417 072416  001232  001234  041346  DT111:  .WORD   $TMP0,$TMP1,$TAB,$TMP2,0
6418         072416                  DT112=DT111
```

```
 6419           072416                    DT113=DT111
 6420           072416                    DT114=DT111
 6421           072416                    DT115=DT111
 6422           072416                    DT116=DT111
 6423           070662                    DT117=DT3
 6424           070662                    DT120=DT3
 6425           072142                    DT121=DT47
 6426           071546                    DT122=DT34
 6427           072174                    DT123=DT51
 6428 072430   001232  001234  041346    DT124:  .WORD   $TMP0,$TMP1,$TAB,$TMP2,$TAB,$TMP3,$TMP4,$CRLF
 6429 072450   042631  001313  042647            .WORD   MS6,$CRLF,MS7,$TMP5,MS10,$TMP6,$CRLF
 6430 072466   042676  001313  042647            .WORD   MS11,$CRLF,MS7,$TMP5,MS10,$TMP7,$CRLF,MS37,$CRLF,$TMP10,0
 6431           072430                    DT125=DT124
 6432           072416                    DT126=DT111
 6433           072416                    DT127=DT111
 6434           070662                    DT130=DT3
 6435 072514   001232  001234  041346    DT131:  .WORD   $TMP0,$TMP1,$TAB,$TMP2,$CRLF,MS37,$CRLF,$TMP3
 6436 072534   001313  043623  001313            .WORD   $CRLF,MS40,$CRLF,$TMP4,$CRLF,MS415,$CRLF,$TMP5,0
 6437           072514                    DT132=DT131
 6438 072556   001232  001234  041346    DT133:  .WORD   $TMP0,$TMP1,$TAB,$TMP2,$CRLF,MS41,$CRLF,$TMP3
 6439 072576   001313  043703  001313            .WORD   $CRLF,MS42,$CRLF,$TMP4,$CRLF,MS43,$CRLF,$TMP5
 6440 072616   001313  043736  001313            .WORD   $CRLF,MS44,$CRLF,$TMP6,0
 6441           072556                    DT134=DT133
 6442           072556                    DT135=DT133
 6443           072556                    DT136=DT133
 6444 072630   001232  001234  041346    DT137:  .WORD   $TMP0,$TMP1,$TAB,$TMP2,$TMP10,$TAB,$TMP11,0
 6445           072630                    DT140=DT137
 6446           072630                    DT141=DT137
 6447           072630                    DT142=DT137
 6448           072556                    DT143=DT133
 6449           072556                    DT144=DT133
 6450           072630                    DT145=DT137
 6451           072630                    DT146=DT137
 6452           072556                    DT147=DT133
 6453           072556                    DT150=DT133
 6454           072630                    DT151=DT137
 6455           072556                    DT152=DT133
 6456           072630                    DT153=DT137
 6457 072650   001232  001234  041346    DT154:  .WORD   $TMP0,$TMP1,$TAB,$TMP2,0
 6458           072650                    DT155=DT154
 6459           072556                    DT156=DT133
 6460           072556                    DT157=DT133
 6461           072556                    DT160=DT133
 6462           072556                    DT161=DT133
 6463           072556                    DT162=DT133
 6464           072556                    DT163=DT133
 6465           070662                    DT164=DT3
 6466           070662                    DT165=DT3
 6467           072556                    DT166=DT133
 6468           072556                    DT167=DT133
 6469           072556                    DT170=DT133
 6470           072556                    DT171=DT133
 6471           072556                    DT172=DT133
 6472           072556                    DT173=DT133
 6473           072556                    DT174=DT133
 6474           072556                    DT175=DT133
 6475           072556                    DT176=DT133
```

DATA TABLES

```
 6476           072556                    DT177=DT133
 6477           072556                    DT200=DT133
 6478           072556                    DT201=DT133
 6479           072556                    DT202=DT133
 6480           072556                    DT203=DT133
 6481           072556                    DT204=DT133
 6482           070662                    DT205=DT3
 6483           072556                    DT206=DT133
 6484           072556                    DT207=DT133
 6485           072556                    DT210=DT133
 6486 072662   001232  001234  041346     DT211:  .WORD   $TMP0,$TMP1,$TAB,$TMP2,$TAB,$TMP3,0
 6487 072700   001232  001234  041346     DT212:  .WORD   $TMP0,$TMP1,$TAB,$TMP2,0
 6488           072700                    DT213=DT212
 6489
 6490
 6491
 6492
 6493                                     ;12345
 6494           000001                            .END
```

SYMBOL TABLE

```
AADATO  026546        ADDW12= 000000        A6      004516        BERR1   004752        CCP3    030364
AADONE  026646        ADDW13= 000000        A7      004542        BIT0  = 000001        CCP4    030374
AAERR0  026200        ADDW14= 000000        BBDATO  031670        BIT00 = 000001        CCP5    030404
AAERR1  026266        ADDW15= 000000        BBDONE  032020        BIT01 = 000002        CCP6    030414
AAERR2  026322        ADDW2 = 000000        BBER0   031356        BIT02 = 000004        CCP7    030424
AAERR3  026356        ADDW3 = 000000        BBER1   031416        BIT03 = 000010        CC1     026652
AAERR4  026366        ADDW4 = 000000        BBER10  031376        BIT04 = 000020        CC10    027076
AAERR5  026422        ADDW5 = 000000        BBER11  031432        BIT05 = 000040        CC11    027106
AAERR6  026456        ADDW6 = 000000        BBER2   031454        BIT06 = 000100        CC12    027114
AAERR7  026512        ADDW7 = 000000        BBER3   031472        BIT07 = 000200        CC13    027126
AAER10  026274        ADDW8 = 000000        BBER4   031526        BIT08 = 000400        CC14    027162
AAPAT0  026556        ADDW9 = 000000        BBER40  031542        BIT09 = 001000        CC15    027206
AAPAT1  026566        ADEVCT= 000000        BBER5   031562        BIT1  = 000002        CC16    027226
AAPAT2  026576        ADEVM = 000000        BBER6   031600        BIT10 = 002000        CC17    027236
AAPAT3  026606        ADONE   004660        BBER7   031634        BIT11 = 004000        CC18    027244
AAPAT4  026616        AENV  = 000000        BBER8   031652        BIT12 = 010000        CC19    027256
AAPAT5  026626        AENVM = 000000        BBPAT0  031700        BIT13 = 020000        CC2     026706
AAPAT6  026636        AERFLG  004550        BBPAT1  031710        BIT14 = 040000        CC20    027312
AA1     025604        AERR1   004552        BBPAT2  031720        BIT15 = 100000        CC21    027336
AA10    025724        AERR2   004564        BBPAT3  031730        BIT2  = 000004        CC22    027356
AA11    025760        AERR3   004616        BBPAT4  031740        BIT3  = 000010        CC23    027366
AA12    025762        AFATAL= 000000        BBPAT5  031750        BIT4  = 000020        CC24    027374
AA13    026000        AMADR1= 000000        BBPAT6  031760        BIT5  = 000040        CC25    027406
AA14    026020        AMADR2= 000000        BBP10   032000        BIT6  = 000100        CC26    027442
AA15    026040        AMADR3= 000000        BBP11   032010        BIT7  = 000200        CC27    027464
AA16    026046        AMADR4= 000000        BBP7    031770        BIT8  = 000400        CC28    027504
AA17    026052        AMAMS1= 000000        BB1     030470        BIT9  = 001000        CC29    027514
AA2     025646        AMAMS2= 000000        BB10    030654        BPTVEC= 000014        CC3     026730
AA20    026056        AMAMS3= 000000        BB11    030674        B1      004676        CC30    027522
AA21    026060        AMAMS4= 000000        BB12    030704        B2      004700        CC31    027534
AA22    026116        AMSGAD= 000000        BB13    030712        B3      004716        CC32    027570
AA23    026120        AMSGLG= 000000        BB14    030724        CCDATO  030324        CC33    027612
AA24    026140        AMSGTY= 000000        BB15    030760        CCDONE  030464        CC34    027632
AA25    026160        AMTYP1= 000000        BB16    031004        CCER0   027666        CC35    027642
AA26    026166        AMTYP2= 000000        BB17    031014        CCER1   027722        CC36    027650
AA27    026172        AMTYP3= 000000        BB2     030532        CCER10  030234        CC37    027662
AA3     025650        AMTYP4= 000000        BB20    031026        CCER11  030252        CC4     026750
AA4     025666        APASS = 000000        BB21    031062        CCER12  030270        CC5     026760
AA5     025706        APRIOR= 000000        BB22    031106        CCER13  030306        CC6     026766
AA6     025714        APTCSU= 000040        BB23    031126        CCER2   027760        CC7     027000
AA7     025722        APTENV= 000001        BB24    031136        CCER22  027774        CC8     027034
ABASE = 000000        APTSIZ= 000200        BB25    031144        CCER3   030016        CC9     027056
ACDW1 = 000000        APTSPO= 000100        BB26    031156        CCER4   030034        CDONE   005732
ACDW2 = 000000        ASWREG= 000000        BB27    031212        CCER44  030050        CERR1   005466
ACPUOP= 000000        ATESTN= 000000        BB3     030536        CCER5   030072        CERR2   005564
AC0   =%000000        AUNIT = 000000        BB30    031234        CCER50  027736        CERR3   005664
AC1   =%000001        AUSWR = 000000        BB31    031244        CCER55  030106        CERR4   005706
AC2   =%000002        AVECT1= 000000        BB32    031256        CCER6   030126        CFCC1   042277
AC3   =%000003        AVECT2= 000000        BB33    031312        CCER7   030162        CKSWR = 104407
AC4   =%000004        A05     004466        BB34    031334        CCER8   030200        CNT   = 000213
AC5   =%000005        A1      004340        BB35    031344        CCER90  027704        CPC     005730
AC6   =%000006        A11     004340        BB4     030554        CCP0    030334        CPSPUR  041174
AC7   =%000007        A12     004354        BB5     030564        CCP1    030344        CPTWO   041212
ADDW0 = 000000        A2      004412        BB6     030576        CCP10   030434        CR    = 000015
ADDW1 = 000000        A3      004416        BB7     030632        CCP11   030444        CRLF  = 000200
ADDW10= 000000        A4      004464        BDONE   004766        CCP12   030454        C1      005006
ADDW11= 000000        A5      004470        BERR    004722        CCP2    030354        C15     005034
```

SYMBOL TABLE

```
C2      005060          DD24    032572          DF133   070524          DF23    067713          DH105 = 064553
C25     005074          DD25    032612          DF134 = 070524          DF24    067723          DH106 = 065445
C3      005122          DD26    032622          DF135 = 070524          DF25    067732          DH107 = 065403
C35     005152          DD27    032630          DF136 = 070524          DF26    067756          DH11  = 064553
C4      005200          DD3     032102          DF137   070550          DF27    070003          DH110 = 066055
C45     005212          DD30    032642          DF14  = 067574          DF3     067564          DH111 = 065403
C5      005240          DD31    032676          DF140 = 070550          DF30    070023          DH112   066253
C55     005266          DD32    032720          DF141 = 070550          DF31  = 067756          DH113 = 065403
C6      005312          DD33    032740          DF142 = 070550          DF32  = 070003          DH114 = 066253
C65     005326          DD34    032750          DF143 = 070524          DF33    070037          DH115   066472
C7      005354          DD35    032756          DF144 = 070524          DF34    070065          DH116 = 066472
C75     005404          DD36    032774          DF145 = 070550          DF35  = 070065          DH117 = 064673
C8      005432          DD37    033030          DF146 = 070550          DF36  = 070065          DH12  = 000000
C85     005444          DD38    033052          DF147 = 070524          DF37  = 070065          DH120   066756
DDDATO  033720          DD39    033072          DF15  = 067574          DF4   = 067564          DH121 = 064553
DDDONE  034050          DD4     032122          DF150 = 070524          DF40    070111          DH122 = 066003
DDER0   033132          DD40    033102          DF151 = 070550          DF41    070137          DH123 = 066003
DDER1   033150          DD41    033110          DF152 = 070524          DF42    070153          DH124 = 064673
DDER10  033566          DD42    033126          DF153 = 070550          DF43  = 070153          DH125 = 064673
DDER11  033624          DD5     032132          DF154   070560          DF44    070201          DH126 = 064553
DDER12  033662          DD6     032140          DF155 = 070560          DF45    070224          DH127 = 066055
DDER2   033206          DD7     032156          DF156 = 070524          DF46    070251          DH13  = 000000
DDER3   033244          DD8     032212          DF157 = 070524          DF47    070303          DH130 = 066756
DDER4   033302          DERR1   006060          DF16    067656          DF5     067574          DH131 = 066055
DDER5   033340          DERR2   006154          DF160 = 070524          DF50    070320          DH132 = 066055
DDER6   033376          DF1     067544          DF161   070564          DF51    070344          DH133   067046
DDER7   033434          DF10  = 067574          DF162 = 070524          DF52  = 070303          DH134 = 067046
DDER8   033472          DF100 = 070371          DF163 = 070564          DF53    070362          DH135 = 067046
DDER9   033530          DF101 = 070416          DF164 = 067564          DF54    070371          DH136 = 067046
DDISP = 177570          DF102 = 070440          DF165 = 067564          DF55  = 070303          DH137   067156
DDONE   006200          DF103 = 070416          DF166 = 070524          DF56  = 070362          DH14  = 064553
DDP0    033730          DF104 = 070371          DF167 = 070524          DF57  = 070371          DH140 = 067156
DDP1    033740          DF105 = 070416          DF17  = 067564          DF6   = 067574          DH141 = 067156
DDP2    033750          DF106 = 070440          DF170 = 070564          DF60  = 070362          DH142 = 067156
DDP3    033760          DF107 = 070416          DF171 = 070564          DF61  = 070371          DH143 = 067046
DDP4    033770          DF11  = 067574          DF172 = 070524          DF62  = 070371          DH144 = 067046
DDP5    034000          DF110 = 070371          DF173 = 070524          DF63  = 070371          DH145 = 067156
DDP6    034010          DF111   070447          DF174 = 070564          DF64    070407          DH146 = 067156
DDP7    034020          DF112 = 070447          DF175 = 070564          DF65  = 070407          DH147 = 066055
DDP8    034030          DF113 = 070447          DF176 = 070524          DF66  = 070371          DH15  = 064553
DDP9    034040          DF114 = 070447          DF177 = 070524          DF67  = 067675          DH150   067346
DD1     032024          DF115 = 070447          DF2     067554          DF7   = 067574          DH151 = 067156
DD10    032234          DF116 = 070447          DF20    067665          DF70    070416          DH152 = 067046
DD11    032244          DF117 = 067564          DF200 = 070524          DF71  = 070416          DH153 = 067156
DD12    032262          DF12    067610          DF201 = 070524          DF72    070440          DH154   067437
DD13    032316          DF120 = 067564          DF202 = 070524          DF73  = 070371          DH155 = 067437
DD14    032340          DF121 = 070303          DF203 = 070524          DF74  = 067675          DH156 = 066055
DD15    032350          DF122 = 070320          DF204 = 070524          DF75  = 070416          DH157 = 066055
DD16    032366          DF123 = 070344          DF205 = 067564          DF76  = 070416          DH16    064613
DD17    032422          DF124   070453          DF206 = 070524          DF77  = 070440          DH160 = 066055
DD18    032444          DF125 = 070453          DF207 = 070524          DH1     064203          DH161 = 066055
DD19    032464          DF126 = 070447          DF21    067675          DH10  = 064553          DH162 = 066055
DD2     032060          DF127 = 070447          DF210 = 070524          DH100 = 066055          DH163 = 066055
DD20    032474          DF13    067642          DF211   070610          DH101 = 064553          DH164 = 064673
DD21    032502          DF130 = 067564          DF212 = 070610          DH102 = 065445          DH165 = 064673
DD22    032514          DF131   070504          DF213 = 070610          DH103 = 065403          DH166 = 066055
DD23    032550          DF132 = 070504          DF22    067701          DH104 = 066055          DH167 = 066055
```

SYMBOL TABLE

```
DH17    064673          DH6   = 064553          DT136 = 072556          DT26    071320          EDONE   006344
DH170 = 066055          DH60  = 065445          DT137   072630          DT27    071374          EEDATO  034552
DH171 = 066055          DH61  = 066055          DT14  = 070736          DT3     070662          EEDONE  034634
DH172 = 066055          DH62  = 066055          DT140 = 072630          DT30    071436          EEER0   034344
DH173 = 066055          DH63  = 066055          DT141 = 072630          DT31  = 071320          EEER1   034362
DH174 = 066055          DH64    066165          DT142 = 072630          DT32  = 071374          EEER2   034420
DH175 = 066055          DH65    066116          DT143 = 072556          DT33    071470          EEER3   034456
DH176 = 066055          DH66  = 066055          DT144 = 072556          DT34    071546          EEER4   034514
DH177 = 066055          DH67  = 064553          DT145 = 072630          DT35  = 071546          EEP0    034562
DH2     064273          DH7   = 064553          DT146 = 072630          DT36  = 071546          EEP1    034574
DH20    064763          DH70  = 064553          DT147 = 072556          DT37  = 071546          EEP2    034604
DH200 = 066055          DH71  = 065403          DT15  = 070736          DT4   = 070662          EEP3    034614
DH201 = 066055          DH72  = 065445          DT150 = 072556          DT40  = 071470          EEP4    034624
DH202 = 066055          DH73  = 066055          DT151 = 072630          DT41    071620          EERR0   006262
DH203 = 066055          DH74  = 064553          DT152 = 072556          DT42    071652          EERR1   006300
DH204 = 066055          DH75  = 064553          DT153 = 072630          DT43  = 071652          EERR2   006314
DH205 = 064673          DH76  = 065403          DT154   072650          DT44    071730          EE1     034054
DH206 = 066055          DH77  = 065445          DT155 = 072650          DT45    072000          EE10    034304
DH207 = 066055          DISPLA  001142          DT156 = 072556          DT46    072054          EE11    034314
DH21  = 064553          DISPRE  000174          DT157 = 072556          DT47    072142          EE12    034322
DH210 = 066055          DPAT3   016642          DT16    071100          DT5     070704          EE13    034340
DH211   067477          DSWR  = 177570          DT160 = 072556          DT50  = 071546          EE2     034110
DH212 = 064553          DT1     070616          DT161 = 072556          DT51    072174          EE3     034132
DH213 = 064553          DT10  = 070736          DT162 = 072556          DT52  = 072142          EE4     034152
DH22    065051          DT100 = 072252          DT163 = 072556          DT53    072232          EE5     034162
DH23    065106          DT101 = 072330          DT164 = 070662          DT54    072252          EE6     034170
DH24    065244          DT102 = 072330          DT165 = 070662          DT55  = 072142          EE7     034206
DH25    065403          DT103 = 072330          DT166 = 072556          DT56  = 072232          EE8     034242
DH26    065445          DT104 = 072252          DT167 = 072556          DT57  = 072252          EE9     034264
DH27  = 065445          DT105 = 072330          DT17  = 070662          DT6     070736          EMTVEC= 000030
DH3     064366          DT106 = 072376          DT170 = 072556          DT60  = 072232          EM1     043760
DH30  = 000000          DT107 = 072330          DT171 = 072556          DT61  = 072252          EM10  = 044166
DH31  = 065445          DT11  = 070736          DT172 = 072556          DT62  = 072252          EM100   053070
DH32  = 065445          DT110 = 072252          DT173 = 072556          DT63  = 072252          EM101   053130
DH33    065533          DT111   072416          DT174 = 072556          DT64    072310          EM102   053254
DH34  = 065533          DT112 = 072416          DT175 = 072556          DT65  = 072310          EM103   053326
DH35  = 065533          DT113 = 072416          DT176 = 072556          DT66  = 072252          EM104   053431
DH36  = 065533          DT114 = 072416          DT177 = 072556          DT67  = 071142          EM105   053472
DH37  = 065533          DT115 = 072416          DT2     070640          DT7   = 070736          EM106   053617
DH4     064457          DT116 = 072416          DT20    071120          DT70    072330          EM107   053672
DH40  = 065533          DT117 = 070662          DT200 = 072556          DT71  = 072330          EM11    044307
DH41  = 000000          DT12    070760          DT201 = 072556          DT72    072376          EM110   053776
DH42    065636          DT120 = 070662          DT202 = 072556          DT73  = 072252          EM111   054040
DH43  = 065636          DT121 = 072142          DT203 = 072556          DT74  = 071142          EM112 = 054040
DH44  = 000000          DT122 = 071546          DT204 = 072556          DT75  = 072330          EM113   054142
DH45    065740          DT123 = 072174          DT205 = 070662          DT76  = 072330          EM114 = 054142
DH46    065757          DT124   072430          DT206 = 072556          DT77  = 072376          EM115 = 054040
DH47  = 065740          DT125 = 072430          DT207 = 072556          D1      005764          EM116 = 054142
DH5     064553          DT126 = 072416          DT21    071142          D10     006166          EM117   054244
DH50    066003          DT127 = 072416          DT210 = 072556          D2      006010          EM12  = 000000
DH51  = 066003          DT13    071046          DT211   072662          D3      006012          EM120   054400
DH52  = 064553          DT130 = 070662          DT212   072700          D4      006016          EM121   054534
DH53  = 065445          DT131   072514          DT213 = 072700          D5      006030          EM122   054653
DH54    066055          DT132 = 072514          DT22    071154          D6      006044          EM123   054752
DH55  = 064553          DT133   072556          DT23    071202          D7      006054          EM124   055013
DH56  = 065445          DT134 = 072556          DT24    071224          D8      006124          EM125   055106
DH57  = 066055          DT135 = 072556          DT25    071250          D9      006136          EM126   055176
```

SYMBOL TABLE

```
EM127   055405        EM210   064126        ERRVEC= 000004        FF2     034674        GPAT02  014304
EM13  = 000000        EM211 = 044166        ERTYPE  040510        FF3     034716        GPAT03  014306
EM130   055620        EM212 = 044222        ERT1    040672        FF4     034724        GPAT10  014310
EM131   055720        EM213 = 044254        ERT2    041110        FF5     034734        GPAT11  014312
EM132   055[illegible]        EM22    045451        ERT3    041114        FF6     034770        GPAT12  014314
EM133   056020        EM23  = 045451        ERT4    041124        FF7     035012        GPAT13  014316
EM134   056057        EM24  = 045451        ERT5    041136        FPSMS   041465        GRESET  014040
EM135   056116        EM25    045536        E1      006220        FPSPUR  041142        GSETUP  013762
EM136   056155        EM26    045651        E2      006234        FPVECT= 000244        GSUM    014172
EM137 = 056020        EM27  = 045651        E3      006234        FXDAT0  010230        GS1     014012
EM14    044370        EM3     044061        E4      006236        FXDAT1  010232        GTSWR = 104406
EM140 = 056057        EM30    045717        FDATI0  010164        FXDAT2  010234        G1      012010
EM141 = 056116        EM31    045771        FDATI1  010166        FXDAT3  010236        G10     012316
EM142 = 056155        EM32  = 045771        FDATI2  010170        FXDAT4  010240        G11     012320
EM143   056214        EM33    046037        FDATI3  010172        FXDAT5  010242        G12     012412
EM144   056247        EM34    046100        FDATI4  010174        FXDAT6  010244        G13     012444
EM145 = 056214        EM35    046202        FDATI5  010176        FXDAT7  010246        G14     012446
EM146 = 056247        EM36    046304        FDATI6  010200        F1      006350        G15     012540
EM147   056302        EM37    046405        FDATI7  010202        F10     006572        G16     012572
EM15    044513        EM4     044126        FDAT00  010206        F11     006574        G17     012574
EM150 = 056302        EM40    046506        FDAT01  010210        F12     006612        G2      012042
EM151 = 056302        EM41    046657        FDAT02  010212        F13     006644        G20     012666
EM152   056334        EM42    046714        FDAT03  010214        F135    006624        G21     012720
EM153 = 056334        EM43    047035        FDAT04  010216        F14     006654        G22     012722
EM154   056366        EM44    047156        FDAT05  010220        F15     006664        G23     013014
EM155   056620        EM45  = 047156        FDAT06  010222        F16     006674        G24     013046
EM156   057053        EM46    047221        FDAT07  010224        F17     006704        G25     013050
EM157   057270        EM47    047277        FDONE   010250        F2      006406        G26     013142
EM16    044636        EM5     044166        FECMS   041531        F20     006714        G27     013174
EM160   057507        EM50    047415        FERR0   006756        F21     006724        G3      012044
EM161   057714        EM51    047513        FERR1   007014        F22     006734        G30     013176
EM162   060121        EM52    047554        FERR10  007574        F23     006752        G31     013270
EM163   060166        EM53    047675        FERR11  007630        F3      006426        G32     013322
EM164   060233        EM54    050072        FERR2   007112        F4      006430        G33     013324
EM165   060300        EM55    050136        FERR20  007650        F5      006446        G34     013416
EM166   060345        EM56    050257        FERR21  007766        F6      006522        G35     013450
EM167   060455        EM57    050454        FERR25  010016        F7      006552        G36     013452
EM17    044707        EM6     044222        FERR26  010134        GADR    014350        G37     013544
EM170   060714        EM60    050520        FERR3   007152        GAND0   014320        G4      012136
EM171   061024        EM61    050715        FERR4   007144        GAND1   014322        G40     013576
EM172   061263        EM62    050761        FERR5   007250        GAND2   014324        G41     013600
EM173   061522        EM63    051153        FERR6   007304        GAND3   014326        G42     013672
EM174   061761        EM64    051345        FERR7   007440        GCMP    014060        G43     013724
EM175   062220        EM65  = 051345        FER2    007116        GDAT00  014340        G44     013726
EM176   062457        EM66    051501        FFDAT0  035142        GDAT01  014342        G5      012170
EM177   062614        EM67    051544        FFDONE  035222        GDAT02  014344        G6      012172
EM2     044015        EM7     044254        FFER0   035032        GDAT03  014346        G7      012264
EM20    045142        EM70    051775        FFER1   035046        GDONE   014352        HADR    015106
EM200   062751        EM71    052120        FFER2   035104        GERR1   014106        HA1R    015162
EM201   063106        EM72    052222        FFP0    035152        GFLAG1  014274        HA1W    015112
EM202   063243        EM73    052276        FFP1    035162        GFLAG2  014276        HA2R    015172
EM203   063400        EM74    052336        FFP2    035172        GOR0    014330        HA2W    015122
EM204   063535        EM75    052567        FFP3    035202        GOR1    014332        HA3R    015202
EM205   063672        EM76    052712        FFP4    035212        GOR2    014334        HA3W    015132
EM206   063737        EM77    053014        FF1     034640        GOR3    014336        HA4R    015212
EM207   064004        ERM10   036232        FF10    035020        GPAT00  014300        HA4W    015142
EM21    045320        ERROR = 104000        FF11    035030        GPAT01  014302        HA5R    015222
```

SYMBOL TABLE

HA5W	015152	I11	010454	KDONE	016644	MNUMBE=	000213	M5	017262
HCLR	015036	I12	010462	KERR0	016466	MNUM0	043143	M6	017266
HCLR1	015046	I13	010476	KERR1	016532	MNUM1	043151	M7	017276
HCMP	015000	I14	010540	KERR2	016556	MNUM2	043156	M8	017306
HCMP1	015020	I15	010542	KPAT0	016634	MNUM3	043163	M9	017316
HCMP2	015030	I16	010544	KPAT1	016636	MNUM4	043172	NDATI0	020220
HDAT1	015232	I17	010562	KPAT2	016640	MNUM5	043200	NDATI1	020222
HDAT2	015242	I2	010300	K1	016360	MPAT10	017472	NDATI2	020224
HDAT3	015252	I20	010622	K10	016506	MPAT11	017474	NDATI3	020226
HDAT4	015262	I21	010636	K2	016404	MPAT12	017476	NDAT00	020156
HDAT5	015272	I22	010642	K3	016406	MPAT13	017500	NDAT01	020160
HDONE	015302	I23	010656	K4	016410	MPAT20	017502	NDAT02	020162
HERROR	015054	I3	010346	K5	016444	MPAT21	017504	NDAT03	020164
HFLAG	015110	I4	010350	K6	016452	MPAT22	017506	NDONE	020230
HSTD	014722	I5	010352	K7	016464	MPAT23	017510	NERR0	017720
HT =	000011	I6	010376	LDATI0	017150	MS1	042507	NERR1	020020
H1	014364	I7	010412	LDATI1	017152	MS10	042663	NERR10	017752
H10	014626	JBUF0	016306	LDATI2	017154	MS11	042676	NERR11	017764
H11	014660	JBUF1	016310	LDATI3	017156	MS12	042723	NERR2	020054
H12	014712	JBUF2	016312	LDAT00	017162	MS13	042761	NERR20	020026
H2	014404	JBUF3	016314	LDAT01	017164	MS14	042776	NERR3	020064
H3	014414	JDATI0	016316	LDAT02	017166	MS15	043064	NERR4	020074
H4	014466	JDATI1	016320	LDAT03	017170	MS16	043111	NERR5	020104
H5	014510	JDATI2	016322	LDONE	017172	MS17	043116	NERR6	020130
H6	014542	JDATI3	016324	LD1	042175	MS2	042521	NOOP1	041547
H7	014574	JDAT00	016326	LD2	042225	MS20	043206	NOOP10	042025
IDATI0	011112	JDAT0	016336	LERR1	017004	MS21	043247	NOOP11	042116
IDATI1	011114	JDAT01	016330	LERR2	017056	MS22	043306	NOOP15	041576
IDATI2	011116	JDAT02	016332	LERR3	017030	MS23	043336	NOOP2	041673
IDATI3	011120	JDAT03	016334	LF =	000012	MS24	043365	NOOP3	041710
IDAT00	011102	JDAT1	016340	LFIEX1	041353	MS25	043460	NOOP4	041722
IDAT01	011104	JDAT2	016342	LFIEX2	041423	MS26	043502	NOOP5	041737
IDATU2	011106	JDAT3	016344	LFPS1	042162	MS27	043517	NOOP6	041765
IDAT03	011110	JDONE	016346	LOOP	004270	MS3	042526	NOOP7	042005
IDONE	011122	JERR0	016166	LPAT10	017126	MS30	043526	NPAT10	020210
IERR0	010662	JERR1	016234	LPAT11	017130	MS31	043535	NPAT11	020212
IERR1	010744	JERR2	016260	LPAT12	017132	MS32	043544	NPAT12	020214
IERR2	010764	J1	016060	LPAT13	017134	MS33	043553	NPAT13	020216
IERR25	011006	J10	016206	LPAT20	017136	MS34	043562	NPAT20	020176
IERR3	011036	J2	016104	LPAT21	017140	MS35	043571	NPAT21	020200
IERR4	011012	J3	016106	LPAT22	017142	MS36	043576	NPAT22	020202
ILLMS	042441	J4	016110	LPAT23	017144	MS37	043605	NPAT23	020204
ILL1	042335	J5	016144	L1	016656	MS4	042555	NULL	041345
ILL2	042400	J6	016152	L2	016716	MS40	043623	N1	017526
IOTVEC=	000020	J7	016164	L3	016720	MS41	043657	N10	017660
IPAT10	011062	KBUF0	016614	L4	016722	MS415	043637	N11	017670
IPAT11	011064	KBUF1	016616	L5	016772	MS42	043703	N12	017672
IPAT12	011066	KBUF2	016620	L6	017000	MS43	043721	N13	017706
IPAT13	011070	KBUF3	016622	MDAT00	017512	MS44	043736	N14	017716
IPAT20	011072	KDATI0	016604	MDAT01	017514	MS5	042567	N2	017552
IPAT21	011074	KDATI1	016606	MDAT02	017516	MS6	042631	N3	017554
IPAT22	011076	KDATI2	016610	MDAT03	017520	MS7	042647	N4	017556
IPAT23	011100	KDATI3	016612	MDONE	017522	M1	017176	N5	017574
I1	010262	KDAT00	016624	MERR0	017346	M15	017216	N6	017604
I10	010444	KDAT01	016626	MERR1	017404	M2	017222	N7	017614
I105	010452	KDAT02	016630	MERR2	017320	M3	017224	N8	017632
I106	010446	KDAT03	016632	MERR3	017432	M4	017226	N9	017644

SYMBOL TABLE

```
ODATI0  020724        PERR14  021160        QERR3   021766        STKLMT= 177774        TERR2   011610
ODATI1  020726        PERR15  021176        QERR4   021774        STST1   042366        TERR25  011624
ODATI2  020730        PERR16  021206        QPAT10  022050        ST1     042251        TERR3   011654
ODATI3  020732        PERR17  021214        QPAT11  022052        ST2     042266        TERR4   011636
ODATO0  020662        PERR2   021324        QPAT12  022054        SWR     001140        TKVEC = 000060
ODATO1  020664        PERR20  021242        QPAT13  022056        SWREG   000176        TPAT10  011700
ODATO2  020666        PERR21  021252        QPAT20  022060        SW0   = 000001        TPAT11  011702
ODATO3  020670        PERR22  021260        QPAT21  022062        SW00  = 000001        TPAT12  011704
ODONE   020734        PIRQ  = 177772        QPAT22  022064        SW01  = 000002        TPAT13  011706
OERR0   020424        PIRQVE= 000240        QPAT23  022066        SW02  = 000004        TPAT20  011710
OERR1   020524        POWERM  041300        Q1      021414        SW03  = 000010        TPAT21  011712
OERR10  020456        PPAT10  021360        Q10     021552        SW04  = 000020        TPAT22  011714
OERR11  020470        PPAT11  021362        Q2      021436        SW05  = 000040        TPAT23  011716
OERR2   020560        PPAT12  021364        Q3    = 021440        SW06  = 000100        TPVEC = 000064
OERR20  020532        PPAT13  021366        Q4      021442        SW07  = 000200        TRAPVE= 000034
OERR3   020570        PROGNU= 000001        Q5      021464        SW08  = 000400        TRTVEC= 000014
OERR4   020600        PR0   = 000000        Q6      021506        SW09  = 001000        TST1    004270
OERR5   020610        PR1   = 000040        Q7      021516        SW1   = 000002        TST10   011124
OERR6   020634        PR2   = 000100        Q8      021530        SW10  = 002000        TST11   011742
OPAT10  020714        PR3   = 000140        Q9      021544        SW11  = 004000        TST12   014354
OPAT11  020716        PR4   = 000200        RDCHR = 104410        SW12  = 010000        TST13   015304
OPAT12  020720        PR5   = 000240        RESREG= 104412        SW13  = 020000        TST14   016050
OPAT13  020722        PR6   = 000300        RESVEC= 000010        SW14  = 040000        TST15   016350
OPAT20  020700        PR7   = 000340        RSETUP= 104413        SW15  = 100000        TST16   016646
OPAT21  020702        PS    = 177776        R6    =%000006        SW2   = 000004        TST17   017174
OPAT22  020704        PSW   = 177776        R7    =%000007        SW3   = 000010        TST2    004662
OPAT23  020706        PWRVEC= 000024        SADR    016022        SW4   = 000020        TST20   017524
OPAT24  020710        P1      020740        SAVREG= 104411        SW5   = 000040        TST21   020232
O1      020234        P2      020762        SCOPE = 000004        SW6   = 000100        TST22   020736
O10     020364        P3    = 020764        SDATO0  016036        SW7   = 000200        TST23   021412
O11     020374        P4      020766        SDATO1  016040        SW8   = 000400        TST24   022112
O12     020376        P5      021010        SDATO2  016042        SW9   = 001000        TST25   023252
O13     020412        P6      021032        SDATO3  016044        S1      015306        TST26   024050
O14     020422        P7      021042        SDONE   016046        S10     015464        TST27   024664
O2      020260        P8      021052        SERR0   015526        S11     015510        TST3    004770
O3      020262        QDATI0  022100        SERR1   015736        S12     015520        TST30   025274
O4      020264        QDATI1  022102        SERR10  015546        S2      015346        TST31   025602
O5      020302        QDATI2  022104        SERR15  015626        S3      015350        TST32   026650
O6      020312        QDATI3  022106        SERR2   015666        S4      015354        TST33   030466
O7      020322        QDATO0  022070        SERR20  015646        S5      015400        TST34   032022
O8      020336        QDATO1  022072        SERR3   015712        S6      015410        TST35   034052
O9      020350        QDATO2  022074        SERR4   015604        S7      015416        TST36   034636
PDATI0  021370        QDATO3  022076        SERR5   015770        S8      015456        TST37   035222
PDATI1  021372        QDONE   022110        SERR6   015700        S9      015460        TST4    005734
PDATI2  021374        QERR0   021556        SERR7   015724        TAB   = 000011        TST5    006202
PDATI3  021376        QERR1   022022        SETD1   042313        TBITVE= 000014        TST6    006346
PDATO0  021400        QERR11  021566        SETF1   042305        TDATI0  011730        TST7    010252
PDATO1  021402        QERR12  021604        SETI1   042321        TDATI1  011732        TYPDS = 104405
PDATO2  021404        QERR13  021622        SETL1   042327        TDATI2  011734        TYPE  = 104401
PDATO3  021406        QERR14  021640        SPACE   041350        TDATI3  011736        TYPOC = 104402
PDONE   021410        QERR15  021656        SPAT10  016026        TDATO0  011720        TYPON = 104404
PERR0   021056        QERR16  021666        SPAT11  016030        TDATO1  011722        TYPOS = 104403
PERR1   021276        QERR17  021674        SPAT12  016032        TDATO2  011724        T1      011126
PERR10  021076        QERR2   021756        SPAT13  016034        TDATO3  011726        T10     011274
PERR11  021106        QERR20  021722        STACK = 001100        TDONE   011740        T105    011302
PERR12  021124        QERR21  021732        START   003606        TERR0   011506        T11     011304
PERR13  021142        QERR22  021740        STFS1   042236        TERR1   011570        T12     011312
```

SYMBOL TABLE

T13	011326	U14'	022642	X1	024052	ZPAT01	025552	$DDW7	001420
T14	011370	U15	022676	X10	024254	ZPAT02	025554	$DDW8	001422
T15	011372	U16	022700	X11	024270	ZPAT03	025556	$DDW9	001424
T16	011374	U2	022230	X12	024330	ZPAT10	025560	$DEVCT	001326
T17	011412	U3	022262	X13	024354	ZPAT11	025562	$DEVM	001374
T2	011152	U4	022332	X14	024362	ZPAT12	025564	$DOAGN	035446
T20	011452	U5	022364	X15	024376	ZPAT13	025566	$DTBL	037314
T21	011466	U6	022434	X16	024436	ZPAT20	025570	$ENDAD	035436
T22	011470	U7	022466	X17	024462	ZPAT21	025572	$ENDCT	035256
T23	011502	WDAP00	024036	X2	024112	ZPAT22	025574	$ENULL	035512
T3	011220	WDAT01	024040	X20	024470	ZPAT23	025576	$ENV	001336
T4	011222	WDAT02	024042	X21	024504	ZTMP1	025534	$ENVM	001337
T5	011224	WDAT03	024044	X3	024136	ZTMP2	025536	$EOP	035222
T6	011250	WDONE	024046	X4	024144	Z1	025316	$EOPCT	035250
T7	011264	WPAT00	024026	X5	024162	Z2	025344	$ERFLG	001103
UDONE	023252	WPAT01	024030	X6	024222	Z3	025370	$ERMAX	001115
UERR0	022716	WPAT02	024032	X7	024246	Z4	025376	$ERROR	035776
UERR1	022742	WPAT03	024034	YDAT00	025232	Z5	025410	$ERRPC	001116
UERR10	022750	WSETUP	023774	YDAT01	025234	Z6	025442	$ERRTB	001442
UERR11	023012	W1	023254	YDAT02	025236	$APTHD	003572	$ERTTL	001112
UERR2	023026	W10	023476	YDAT03	025240	$ATYC	037360	$ESCAP	001304
UERR20	023034	W11	023524	YDONE	025272	$ATY1	037334	$ETABL	001336
UERR21	023076	W12	023560	YERR1	025072	$ATY3	037342	$ETEND	001442
UERR3	023112	W13	023604	YERR2	025134	$ATY4	037352	$FATAL	001320
UERR4	023142	W14	023620	YERR3	025176	$AUTOB	001134	$FFLG	037600
UFLAG	023236	W15	023646	YFLAG	025222	$BASE	001372	$FILLC	001156
UPAT00	023166	W16	023706	YPAT00	025242	$BDADR	001122	$FILLS	001155
UPAT01	023170	W17	023732	YPAT01	025244	$BDDAT	001126	$GDADR	001120
UPAT02	023172	W2	023310	YPAT02	025246	$BELL	001306	$GDDAT	001124
UPAT03	023174	W20	023746	YPAT03	025250	$CDW1	001376	$GET42	035410
UPAT10	023176	W3	023334	YPAT10	025252	$CDW2	001400	$GTSWR	037652
UPAT11	023200	W4	023352	YPAT11	025254	$CHARC	036656	$HD =	000003
UPAT12	023202	W5	023402	YPAT12	025256	$CKSWR	037602	$HIBTS	003572
UPAT13	023204	W6	023442	YPAT13	025260	$CLR.T	035426	$ICNT	001104
UPAT20	023206	W7	023466	YPAT20	025262	$CMTAG	001100	$ILLUP	040502
UPAT21	023210	XAPT11	024644	YPAT21	025264	$CM1 =	000024	$INTAG	001135
UPAT22	023212	XDAT00	024622	YPAT22	025266	$CM2 =	000050	$ITEMB	001114
UPAT23	023214	XDAT01	024624	YPAT23	025270	$CM3 =	000024	$LF	001314
UPAT30	023216	XDAT02	024626	YTMP1	025224	$CM4 =	000024	$LFLG	037577
UPAT31	023220	XDAT03	024630	YTMP2	025226	$CNTLG	040211	$LOOP	035504
UPAT32	023222	XDONE	024662	YTMP3	025230	$CNTLU	040204	$LPADR	001106
UPAT33	023224	XERR1	024506	Y1	024714	$CPUOP	001344	$LPERR	001110
UPAT40	023226	XERR2	024570	Y2	024742	$CRLF	001313	$MADR1	001350
UPAT41	023230	XERR3	024534	Y3	024766	$DBLK	037324	$MADR2	001354
UPAT42	023232	XERR4	024604	Y4	025004	$DDW0	001402	$MADR3	001360
UPAT43	023234	XPAT00	024632	Y5	025044	$DDW1	001404	$MADR4	001364
UROM1	023244	XPAT01	024634	Y6	025046	$DDW10	001426	$MAIL	001316
UROM2	023246	XPAT02	024636	Y7	025062	$DDW11	001430	$MAMS1	001346
UROM3	023250	XPAT03	024640	ZDAT00	025540	$DDW12	001432	$MAMS2	001352
UTMP1	023240	XPAT10	024642	ZDAT01	025542	$DDW13	001434	$MAMS3	001356
UTMP2	023242	XPAT12	024646	ZDAT02	025544	$DDW14	001436	$MAMS4	001362
U0	022130	XPAT13	024650	ZDAT03	025546	$DDW15	001440	$MBADR	003574
U1	022160	XPAT20	024652	ZDONE	025600	$DDW2	001406	$MFLG	037576
U10	022536	XPAT21	024654	ZERR1	025444	$DDW3	001410	$MNEW	040227
U11	022540	XPAT22	024656	ZERR2	025506	$DDW4	001412	$MSGAD	001332
U12	022572	XPAT23	024660	ZFLAG	025532	$DDW5	001414	$MSGLG	001334
U13	022610	XTMP	024620	ZPAT00	025550	$DDW6	001416	$MSGTY	001316

```
CKFPAB0 FP11F FLTG PNT PRT A    MACRO M1113  10-OCT-80 08:51  PAGE 75-11        SEQUENCE 185
SYMBOL TABLE

$MSWR   040216          $REG1   001164          $SAVRE  036234          $TMP11  001254          $TRAP2  040262
$MTYP1  001347          $REG10  001202          $SAVR6  040506          $TMP12  001256          $TRP  = 000014
$MTYP2  001353          $REG11  001204          $SCOPE  035516          $TMP13  001260          $TRPAD  040274
$MTYP3  001357          $REG12  001206          $SETUP= 000137          $TMP14  001262          $TSTM   003576
$MTYP4  001363          $REG13  001210          $STUP = 177777          $TMP15  001264          $TSTNM  001102
$MXCNT  035774          $REG14  001212          $SVLAD  035724          $TMP16  001266          $TYPDS  037110
$NULL   001154          $REG15  001214          $SVPC = 003572          $TMP17  001270          $TYPE   036330
$NWTST= 000001          $REG16  001216          $SWR  = 177400          $TMP2   001236          $TYPEC  036542
$OCNT   037104          $REG17  001220          $SWREG  001340          $TMP20  001272          $TYPEX  036660
$OMODE  037106          $REG2   001166          $SWRMK= 000000          $TMP21  001274          $TYPOC  036706
$OVER   035760          $REG20  001222          $SWRMS= 000200          $TMP22  001276          $TYPON  036722
$PASS   001324          $REG21  001224          $TAB    041346          $TMP23  001300          $TYPOS  036662
$PASTM  003600          $REG22  001226          $TBIT   035510          $TMP3   001240          $UNIT   001330
$PWRAD  040464          $REG23  001230          $TERM = 000030          $TMP4   001242          $UNITM  003602
$PWRDN  040324          $REG3   001170          $TESTN  001322          $TMP5   001244          $USWR   001342
$PWRMG  040460          $REG4   001172          $THE    041542          $TMP6   001246          $VECT1  001366
$PWRUP  040376          $REG5   001174          $TIMES  001302          $TMP7   001250          $VECT2  001370
$QUES   001312          $REG6   001176          $TKB    001146          $TN   = 000037          $XTSTR  035530
$RDCHR  040064          $REG7   001200          $TKS    001144          $TPB    001152          $$GET4= 000001
$RDSZ = 000001          $RESRE  036272          $TMP0   001232          $TPFLG  001157          $OFILL  037105
$REGAD  001160          $RTNAD  035506          $TMP1   001234          $TPS    001150          .RSET   041230
$REG0   001162          $RTRN   035502          $TMP10  001252          $TRAP   040240          .$X   = 003572

. ABS.  072712      000
        000000      001
ERRORS DETECTED:  0


VIRTUAL MEMORY USED:  56776 WORDS  ( 222 PAGES)
DYNAMIC MEMORY:  20346 WORDS  ( 78 PAGES)
ELAPSED TIME:  00:06:17
CKFPAB.BIN,CKFPAB.SEQ/-SP/NL:TOC=CKFPAB.MLB/ML,CKFPAB.P11
```