11/24

11/24 CP CLUSTER DIAG AH-F608B-MC
CJKDEBO FICHE 1 OF 3

JAN 1983

digital

11/24

11/24 CP CLUSTER DIAG AH-F608B-MC
CJKDEBO FICHE 2 OF 3

JAN 1983

digital

11/24 CP CLUSTER DIAG AH-F608B-MC
CJKDEBO FICHE 3 OF 3

```
.REM_




















                      IDENTIFICATION
                      --------------



       PRODUCT CODE:  AC-F605B-MC

       PRODUCT NAME:  CJKDEB0 11/24 CP CLUSTER DIAG

       PRODUCT DATE:  AUGUST 1982

       MAINTAINER:  DIAGNOSTIC ENGINEERING




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

NO RESPONSIBILITY IS ASSUMED FOR THE USE OR RELIABILITY OF SOFTWARE ON
EQUIPMENT THAT IS NOT SUPPLIED BY DIGITAL OR ITS AFFILIATED COMPANIES.

COPYRIGHT (C):  1981,1982 BY DIGITAL EQUIPMENT CORPORATION

THE FOLLOWING ARE TRADEMARKS OF DIGITAL EQUIPMENT CORPORATION:

DIGITAL        PDP          UNIBUS         MASSBUS
DEC            DECUS        DECTAPE        DEC/X11
```

```
                                        HISTORY
                                        -------

               REVISION A               FIRST RELEASE OF DIAGNOSTIC
               REVISION B               UPDATED FOR VOLTAGE MONITOR BIT CHECK,AND
                                        FOR KNOWN BUGS REPORTED BY MANUFACTURING.
```

TABLE OF CONTENTS

1.0 GENERAL PROGRAM INFORMATION
1.1 ABSTRACT
1.2 SYSTEM REQUIREMENTS
1.3 RELATED DOCUMENTS AND STANDARDS
1.4 DIAGNOSTIC HIERARCHY PREREQUISITES
1.5 ASSUMPTIONS

2.0 OPERATING INSTRUCTIONS
2.1 LOADING AND STARTING PROCEDURE
2.2 PROGRAM OPTIONS
2.3 EXECUTION TIMES

3.0 ERROR INFORMATION
3.1 ERROR REPORTING PROCEDURES
3.2 ERROR HALTS

4.0 PERFORMANCE AND PROGESS REPORTS
4.1 PERFORMANCE REPORTS
4.2 PROGRESS REPORTS

5.0 DEVICE INFORMATION TABLES

6.0 PROGRAM DESCRIPTION
6.1 PROGRAM EXECUTION CHARACTERISTICS
6.2 SUBTEST SUMMARIES
6.3 SPECIAL SUBROUTINE DESCRIPTION

7.0 LISTING

```
1.0  ABSTRACT

     THIS PROGRAM IS A GO-NOGO TEST FOR THE 11/24 CPU BOARD.  IT TESTS
     THE  CPU INCLUDING EIS, THE MMU, THE FPP, THE LTC AND BOTH SLU'S.
     IT DOES NOT CONTAIN THE CAPABILITIES OF SCOPE LOOPING, ERROR  RE-
     COVERY OR PRINTING OF ERROR INFORMATION.  ERROR HALTS DO INDICATE
     WHICH DEVICE FAILED TO ALLOW THE TECHNICIAN  TO  DETERMINE  WHICH
     DIAGNOSTIC TO USE TO FIX THE BOARD OR WHAT FIELD REPLACEABLE UNIT
     (FRU) MAY FIX THE BOARD.  THE PROGRAM WILL RUN UNDER THE ACT  AND
     APT MANUFACTURING SYSTEMS AND IS CHAINABLE UNDER XXDP.


1.1  SYSTEM REQUIREMENTS

     A.  HARDWARE REQUIREMENTS

         - CPU BASED ON DCF11-AA CHIP SET WITH KTF11-AA AND KEF11-AA CHIPS
           PRESENT.

         - LTC

         - 2 SLU'S COMPATIBLE WITH 11/24 SLU SPECIFICATION.

         - 28K OF MEMORY

         - THE SECOND SLU MUST HAVE TURN AROUND CONNECTOR.

     B.  SOFTWARE ENVIRONMENTS

         - APT (MULTI-CPU TESTER)

         - ACT

         - XXDP (SLIDE)

         - STAND-ALONE


1.2  RELATED DOCUMENTS AND STANDARDS

         - ASSEMBLED WITH SYSMAC;  SEE FIRST PAGE OF LISTING FOR REVI-
           SION NUMBER.

         - M7133 MODULE SPECIFICATION

         - DIAGNOSTIC ENGINEERING FUNCTIONAL SPECIFICATION FOR SPECIAL
           MANUFACTURING TEST BGI-79-003-00-U.

         - DIAGNOSTIC ENGINEERING PROJECT  PLAN  FOR  11/24)
           BGI-78-002-02-U.
```

- DIAGNOSTIC ENGINEERING STANDARDS AND CONVENTIONS 175-003-009-02.

1.3 PREREQUISITE DIAGNOSTICS

NONE

1.4 ASSUMPTIONS

THIS PROGRAM ASSUMES THE MACHINE IS UP SUFFICIENTLY TO ALLOW PROPER OPERATION OF THE MICRO-ODT OF THE DCF11-AA CHIP SET.

2.0 OPERATING INSTRUCTIONS

2.1 LOADING AND STARTING PROCEDURES

TO LOAD AND START THIS PROGRAM USE THE STANDARD PROCEDURES FOR THE DIAGNOSTIC SOFTWARE ENVIRONMENT THAT IS BEING USED.

2.2 PROGRAM OPTIONS

THIS PROGRAM USES THE SOFTWARE SWITCH LOCATION 176 IF PROGRAM IS NOT BEING RUN UNDER APT MODE (BIT 0 SET OF LOCATION $ENV). IF PROGRAM IS BEING RUN IN APT MODE THE LOCATION $SWREG IN THE APT ETABLE IS USED TO STORE OPERATING SWITCHES.

BIT #	DEFINITION
15-6	NOT USED
5	0 - PROGRAM RESERVED -- PROGRAM WILL CLEAR IF CIS CHIP SET NOT ON BOARD
	1 - PROGRAM RESERVED -- PROGRAM WILL SET IF CIS CHIP SET IS ON BOARD
4	0 - TEST SLU2 OF 11/24
	1 - INHIBIT TESTING OF SLU2
3	0 - TEST LTC OF 11/24
	1 - INHIBIT TESTING OF LTC
2	0 - TEST SLU1 OF 11/24
	1 - INHIBIT TESTING OF SLU1
1	0 - TEST FPP INSTRUCTION SET
	1 - INHIBIT TESTING OF FPP
0	0 - TEST MEMORY MANAGEMENT UNIT
	1 - INHIBIT TESTING OF MMEMORY MANAGEMENT UNIT

2.3 EXECUTION TIMES

FIRST PASS RUNTIME (WORST CASE)............45 SEC
LONGEST TEST TIME..........................30 SEC
ADDITIONAL RUNTIME (EXTRA UNITS)...........NONE
LONGEST PASS TIME..........................45 SEC

3.0 ERROR INFORMATION

3.1 ERROR REPORTING PROCEDURES

THE PROGRAM DOES NOT TYPE OUT ANY ERROR REPORTS OF ITS OWN BUT TAKES ADVANTAGE OF THE HARDWARE FEATURE THAT TYPES THE PC WHEN A HALT OCCURS. WHEN AN ERROR IS DETECTED THE PROGRAM JUMPS TO ONE OF SEVEN HALT ROUTINES. THE ROUTINES SIMPLY MOVE A FATAL ERROR NUMBER INTO LOCATION $FATAL, SET THE FATAL ERROR FLAG IN LOCATION $MSGTY AND EITHER HALT OR IF ON APT DO A BRANCH DOT. THE OPERATOR HAS THREE WAYS TO DETERMINE THE FAILING DEVICE; 1) BY EXAMINING LOCATION $FATAL, 2) BY DETERMINING THE HALT ADDRESS AND LOOKING UP THE ADDRESS IN THE LISTING AND 3) BY EXAMINING LOCATION $TESTN WHICH WILL CONTAIN THE TEST NUMBER BEING EXECUTED.

3.2 ERROR HALTS

FOR DISCUSSION SEE SECTION 3.1. THE LABELS FOR THE HALTS AND THE DEVICE THEY INDICATE HAVING FAILED ARE:

CPUHLT: CPU
MMUHLT: MMU
FPPHLT: FPP
LTCHLT: LTC
SL1HLT: SLU1
SL2HLT: SLU2
COMHLT: SYSTEM INTERACTION

*** IF THE LOCATION $FATAL = 177 THEN THE POWER FAILED ***

UPON RECEIVING THE ERROR HALT ADDRESS (PC) FROM THE MICRO-ODT THE OPERATOR CAN LOOK UP THESE TAGS IN THE SYMBOL TABLE AT THE END OF THE LISTING TO DETERMINE WHICH HALT WAS EXECUTED. NOTE: THE PC SUPPLIED BY THE MICRO-ODT WILL BE THE HALT ADDRESS PLUS 2.

THE OPERATOR CAN DETERMINE WHICH TEST THE ERROR OCCURRED IN BY EXAMINING LOCATION "$TESTN". THE TEST NUMBERS EQUATE TO FAILING DEVICES AS FOLLOWS:

DEVICE CONTENTS OF $TESTN

```
            CPU            000001
            MMU            000002
            FPP            000003
            SLU1           000004
            LTC            000005
            SLU2           000006
            SYSTEM
            INTERACTION    000007


4.0   PERFORMANCE AND PROGRESS REPORTS

      THE ONLY REPORT TYPED BY THIS PROGRAM IS  THE  END  PASS  MESSAGE
      WHICH IS:

                CJKDEB END OF PASS #XXX

      WHERE XXX IS THE DECIMAL NUMBER OF PASSES COMPLETED.
```

```
5.0  DEVICE INFORMATION TABLES

SLU1 RCSR

      15  14  13  12  11  10   9   8   7   6   5   4   3   2   1   0
     ---------------------------------------------------------------
     I                              I   I   I                      I
     ---------------------------------------------------------------
                                      I   I
                                      I   I
 RECEIVER DONE ------------------------   I
 RECEIVER INTERRUPT ENABLE ----------------


SLU1 RBUF

      15  14  13  12  11  10   9   8   7   6   5   4   3   2   1   0
     ---------------------------------------------------------------
    I   I   I   I   I               I   I   I   I   I   I   I   I   I
     ---------------------------------------------------------------
      I   I   I   I                                   I
      I   I   I   I                                   I
ERROR-    I   I   I                                   I
OVERRUN ---   I   I                                   I
FRAME ERROR ---   I                                   I
RECEIVE PARITY ----                                   I
ERROR                                                 I
RECEIVED DATA BITS (8) --------------------------------


SLU1 XCSR

      15  14  13  12  11  10   9   8   7   6   5   4   3   2   1   0
     ---------------------------------------------------------------
    I                           I   I   I           I   I           I
     ---------------------------------------------------------------
                                  I   I               I
                                  I   I               I
TRANSMITTER READY -----------------   I               I
TRANSMITTER INTERRUPT ENABLE ----------               I
MAINTENANCE -------------------------------------------
```

```


SLU1 XBUF

      15  14  13  12  11  10  9   8   7   6   5   4   3   2   1   0
     ----------------------------------------------------------------
    I                               I   I   I   I   I   I   I   I   I
     ----------------------------------------------------------------
                                            I
                                            I
TRANSMITTER DATA BITS (8) -------------------



LTC CSR

      15  14  13  12  11  10  9   8   7   6   5   4   3   2   1   0
     ----------------------------------------------------------------
    I                           I   I   I                           I
     ----------------------------------------------------------------
                                  I   I
                                  I   I
LINE CLOCK MONITOR ----------------   I
INTERRUPT ENABLE ---------------------



SLU2 RCSR

      15  14  13  12  11  10  9   8   7   6   5   4   3   2   1   0
     ----------------------------------------------------------------
    I                           I   I   I                           I
     ----------------------------------------------------------------
                                  I   I
                                  I   I
RECEIVER DONE ---------------------   I
INTERRUPT ENABLE ---------------------



SLU2 RBUF

      15  14  13  12  11  10  9   8   7   6   5   4   3   2   1   0
     ----------------------------------------------------------------
    I   I   I   I   I               I   I   I   I   I   I   I   I   I
     ----------------------------------------------------------------
      I   I   I   I                                 I
      I   I   I   I                                 I
ERROR -   I   I   I                                 I
OVERRUN ---   I   I                                 I
FRAME ERROR ---   I                                 I
RECEIVER PARITY ---                                 I
ERROR                                               I
RECEIVER DATA BITS (8) ------------------------------
```

```
SLU2 XCSR

      15  14  13  12  11  10  9   8   7   6   5   4   3   2   1   0
      ---------------------------------------------------------------
                                I   I   I   I                   I   I
      ---------------------------------------------------------------
                                    I   I                       I
                                    I   I                       I
TRANSMITTER DONE -------------------    I                       I
INTERRUPT ENABLE -----------------------                        I
BREAK ----------------------------------------------------------


SLU2 XBUF
      15  14  13  12  11  10  9   8   7   6   5   4   3   2   1   0
      ---------------------------------------------------------------
     I                              I   I   I   I   I   I   I   I   I
      ---------------------------------------------------------------
                                                I
                                                I
TRANSMITTER DATA BITS (8) ----------------------
```

6.0 PROGRAM DESCRIPTION

6.1 PROGRAM EXECUTION CHARACTERISTICS

THIS PROGRAM RUNS THE SAME UNDER ALL DIAGNOSTIC MONITORS. WHEN THE TEST IS STARTED AT ADDRESS 200 OCTAL THE TESTING IS DONE AND ON COMPLETION THE TITLE IS TYPED AS PART OF THE END OF PASS MES-SAGE.

6.2 SUB-TEST SUMMARIES

6.2.1 CENTRAL PROCESSING UNIT SUBTEST -

THESE TESTS CHECK THE BASIC INSTRUCTION SET AND ADDRESSING MODES, THE EXTENDED ELEVEN INSTRUCTION SET (EIS) AND TRAPS TESTING. IT IS EQUIVALENT TO CJKDB.

6.2.2 MEMORY MANAGEMENT UNIT SUBTEST -

THESE TESTS ARE THE SAME AS IN CJKDA, THE KEF11-AA TEST. THE PROGRAM BEGINS BY TESTING SOME OF THE INTERNAL CPU DATA AND ADDRESS PATHS AND ADDRESS DETECTION LOGIC. NEXT THE MEMORY MANAGEMENT REGISTERS ARE CHECKED FOR DATA RELIABILITY, THEN RELOCATION CAPABILITIES (FORMATION OF A PHYSICAL ADDRESS FROM A VIRTUAL ADDRESS AND ASSOCIATED PAGE DESCRIBTOR (PDR) INFORMATION). FINALLY THE ABORT AND STATUS SEGMENTS OF THE LOGIC ARE CHECKED.

6.2.3 FLOATING POINT PROCESSOR SUBTEST -

THE FLOATING POINT PROCESSOR SUBTEST CHECKS FLOATING POINT REGISTERS FIRST USING A LIMITED NUMBER OF FLOATING POINT INSTRUCTIONS. IT THEN VERIFIES THE REST OF THE FLOATING POINT INSTRUCTION SET USING A NUMBER OF DATA PATTERNS FOR EACH INSTRUCTION.

6.2.4 SERIAL LINE UNIT (SLU1) SUBTEST -

THESE TEST FIRST CHECK THE SLU'S REGISTERS FOR ADDRESSING AND DATA CAPABILITIES. THEN USING THE MAINTENANCE MODE OF THE SLU THEY CHECK THE DATA PATHS AND INTERRUPT LOGIC.

6.2.5 LINE TIME CLOCK (LTC) SUBTEST -

FIRST THE REGISTER IS CHECKED FOR ADDRESSING AND BIT SETTING CAPABILITIES THEN THE INTERRUPT LOGIC IS CHECKED. THERE IS ALSO A REPEATABILITY TEST FOR THE CLOCK.

6.2.6 SERIAL LINE UNIT 2 SUBTEST -

THE TESTING DONE HERE IS THE SAME AS FOR SLU1 (SEE SECTION 6.2.4). THE ONLY DIFFERENCE IS THAT SINCE SLU2 DOES NOT HAVE A MAINTENANCE WRAP-AROUND LIKE SLU1 THE TESTING REQUIRES AN EXTERNAL JUMPER TO BE PRESENT.

6.2.7 BLAST SUBTEST

THIS TEST CHECKS THE ABILITY OF THE 11/24 TO HANDLE SYSTEM INTERACTION. THE CPU HAS TO HANDLE DEVICES AT DIFFERENT PRIORITY LEVELS AND ARBITRATE BETWEEN THEM AND ITS OWN PRIORITY. THE TEST SETS UP ALL DEVICES TO INTERRUPT THEN ENABLES THEM ALL AT ONCE. THE SLU'S TRANSFER DATA UNTIL THEY TRANSFER 400(8) BYTES OR UNTIL ONE SECOND (60 TICKS) OF THE LINE CLOCK HAS BEEN RECEIVED. THE PROGRAM THEN VERIFIES THE NUMBER OF TRANSMTTER INTERRUPTS IS EQUAL TO THE NUMBER OF RECEIVER INTERRUPTS. FINALLY THE DATA TRANSFERRED BY EACH DEVICE IS CHECKED.

6.3 SPECIAL SUBROUTINE DESCRIPTIONS

THE ONLY SPECIAL SUBROUTINES ARE THE ERROR ROUTINES EACH SUBTEST HAS IS OWN. THE ROUTINES SIMPLY SET THE FATAL ERROR FLAG IN THE APT MAILBOX AND EITHER "BRANCH SELF" OR "HALT". THIS CHOICE IS DETERMINED IN THE INITIALIZE PORTION OF THE PROGRAM AND IS A "BRANCH SELF" IF RUNNING UNDER APT OR A "HALT" IF RUNNING UNDER ANY OTHER MONITOR.

7.0 LISTING

```
                                .TITLE  CJKDE-B  11/24 CPU CLUSTER DIAG.
                                .ENABLE ABS
                                .NLIST  CND,MC,MD
                                .LIST   ME
             000240             SCOPE=NOP
             000007             R7=%7
             000006             R6=%6
             177776             PS=177776
             177564             TPS=177564
             177566             TPB=177566
             140000             USRM=140000
             030000             PUSRM=30000
             000006             SP=%6
             000006             R6=%6
             000003             TAB=%3
             000001             LAST=%1
             000005             FIRST=%5
             000002             R2=%2
             000000             HLT=HALT
             000003             TRT=3
             000004             ITRAP5=4
             000004             RTRAP5=4                ;RESERVED INST AND ILLEGAL ADDRESSES
             000014             RTRAP4=14               ;FOR TRACE TRAP
             000030             RTRAP3=30               ;FOR EMULATOR TRAP
             000020             RTRAP2=20               ;FOR IOT TRAP
             000034             RTRAP1=34               ;FOR TRAP INST
             177564             TTCSR=177564
             177560             TRCSR=177560
             177562             TKB=177562
             177566             TPB=177566
             000240             BELL=240
             000240             NOP=240
             177776             STATUS=177776
             000077             TRAPA=77
             000010             RTRAP=10
             004700             ILLA=004700
             000100             ILLB=100
             177776             CC=177776
             001000             KERSTK=STBOT
             000600             USESTK=STBOT-200
             104377             EMTA=104377
             104777             TRAPC=104777
             000001             APTENV=1
                        .SBTTL  ACT11 HOOKS

                        ;***************************************************************************
                        ;HOOKS REQUIRED BY ACT11
             000400             $SVPC=.                 ;SAVE PC
             000046             .=46
     000046  131522             $ENDAD                  ;;1)SET LOC.46 TO ADDRESS OF $ENDAD IN .$EOP
             000052             .=52
     000052  000000             .WORD   0               ;;2)SET LOC.52 TO ZERO
             000400             .=$SVPC                 ;; RESTORE PC
             001000             .=1000
```

```
                          .SBTTL  APT MAILBOX-ETABLE

                          ;**********************************************************************
                          .EVEN
001000                    $MAIL:                       ;;APT MAILBOX
001000  000000            $MSGTY: .WORD   AMSGTY   ;;MESSAGE TYPE CODE
001002  000000            $FATAL: .WORD   AFATAL   ;;FATAL ERROR NUMBER
001004  000000            $TESTN: .WORD   ATESTN   ;;TEST NUMBER
001006  000000            $PASS:  .WORD   APASS    ;;PASS COUNT
001010  000000            $DEVCT: .WORD   ADEVCT   ;;DEVICE COUNT
001012  000000            $UNIT:  .WORD   AUNIT    ;;I/O UNIT NUMBER
001014  000000            $MSGAD: .WORD   AMSGAD   ;;MESSAGE ADDRESS
001016  000000            $MSGLG: .WORD   AMSGLG   ;;MESSAGE LENGTH
001020                    $ETABLE:                 ;;APT ENVIRONMENT TABLE
001020     000            $ENV:   .BYTE   AENV     ;;ENVIRONMENT BYTE
001021     000            $ENVM:  .BYTE   AENVM    ;;ENVIRONMENT MODE BITS
001022  000000            $SWREG: .WORD   ASWREG   ;;APT SWITCH REGISTER
001024  000000            $USWR:  .WORD   AUSWR    ;;USER SWITCHES
001026  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
001030                    $ETEND:
                          .MEXIT
                          .SBTTL  APT PARAMETER BLOCK

                          ;**********************************************************************
                          ;SET LOCATIONS 24 AND 44 AS REQUIRED FOR APT
                          ;**********************************************************************
        001030                    .$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  001030                    $APTHDR  ;;POINT TO APT HEADER BLOCK
        001030                    .=.$X    ;;RESET LOCATION COUNTER
                          ;**********************************************************************
                          ;SETUP APT PARAMETER BLOCK AS DEFINED IN THE APT-PDP11 DIAGNOSTIC
                          ;INTERFACE SPEC.

001030                    $APTHD:
001030  000000            $HIBTS: .WORD   0        ;;TWO HIGH BITS OF 18 BIT MAILBOX ADDR.
001032  001000            $MBADR: .WORD   $MAIL    ;;ADDRESS OF APT MAILBOX (BITS 0-15)
001034  000010            $TSTM:  .WORD   10       ;;RUN TIM OF LONGEST TEST
001036  000025            $PASTM: .WORD   25       ;;RUN TIME IN SECS. OF 1ST PASS ON 1 UNIT (QUICK VERIFY)
001040  000000            $UNITM: .WORD   0        ;;ADDITIONAL RUN TIME (SECS) OF A PASS FOR EACH ADDITIONAL UNIT
001042  000014                    .WORD   $ETEND-$MAIL/2 ;;LENGTH MAILBOX-ETABLE(WORDS)
                          ;**********************************************************************
                          ;SOME POINTERS TO CPU TRAP HANDLERS
                          ;**********************************************************************
        000004                    .=4
000004  021256                    T04
000006  000000                    0
000010  021260                    T010
```

```
000012  000000                                  0
000014  021262                                  T014
000016  000000                                  0
000020  021264                                  T020
000022  000000                                  0
        000030                                  .=30
000030  021266                                  T030
000032  000000                                  0
000034  021270                                  T034
000036  000000                                  0
        000114                                  .=114
000114  021272                                  T0114
000116  000000                                  0
        000244                                  .=244
000244  021274                                  T0244
000246  000000                                  0
000250  021276                                  T0250
000252  000000                                  0

        000172                          .=172
000172  000000                          MTFLAG:         0       ;MULTI-TESTER ACTIVE BIT
000174  000000                          DISPREG:        0       ;SOFTWARE DISPLAY REGISTER
000176  000000                          SWREG:          0       ;SOFTWARE SWITCH REGISTER

                                        ;*************************************************************************
                                        ;DATA TABLE FOR USE IN ADDRESSING MODE TESTS
                                        ;*************************************************************************
        000370                                  .=370
000370  000000  000000  000000                  0,0,0,0,0,0
000376  000000  000000  000000
000404  000001  000001  177777                  1,1,-1
                                        ;*************************************************************************
                                        ;SET UP STARTING ADDRESS
        001000                          .=1000
001000  000000                          STBOT:  .WORD 0                         ;STACK POINTER

        000200                                  .=200
000200  000167  000774                          JMP     START
000204  012706  001000                          MOV     #STBOT,R6               ;SET STACK POINTER
000210  012702  001004                          MOV     #$TESTN,R2              ;SET MAILBOX POINTER
000214  000137                                  JMP     @(PC)+                  ;JUMP TO SUBTEST
000216  000000                                  0                               ;ADDR. OF SUBTEST GOES HERE

        001200                          .=1200
                                        .SBTTL **STARTING OF CPU TEST **
001200  012737  000000  001006          START:  MOV     #0,@#$PASS      ;CLEAR PASS COUNT
001206  012737  131574  000024          RESTRT: MOV     #PWRDN,@#24             ;SET UP FOR POWER FAIL
001214  012706  001000                          MOV     #STBOT,R6       ;SET UP STACK
001220  012737  001250  000004                  MOV     #1$,@#4         ;SET UP FOR TIMEOUT IF NO MULTI TESTER
001226  012737  000340  000006                  MOV     #340,@#6
001234  012737  000002  164000                  MOV     #2,@#164000     ;SET BIT1 FOR MULTI TESTER
001242  012737  000001  000172                  MOV     #1,@#MTFLAG     ;SET FLAG TO INDICATE MULTI-TESTER
001250  012737  021256  000004  1$:             MOV     #T04,@#4                ;SET TRAP CATCHER
001256  012737  000000  000006                  MOV     #0,@#6          ;SET HALT BACK IN LOCATION 6
001264  012706  001000                          MOV     #STBOT,R6       ;INITIALIZE STACK POINTER
001270  012737  000001  001004                  MOV     #1,@#$TESTN     ;SET TEST NUMBER TO 1
```

```
001276  012737  000000  001002          MOV     #0,@#$FATAL      ;CLEAR ERROR INDICATOR
001304  012737  000000  001000          MOV     #0,@#$MSGTY      ;CLEAR MESSAGE TYPE(FOR APT)
001312  105737  001020                  TSTB    @#$ENV           ;RUNNING ON APT
001316  001025                          BNE     TS1              ;IF YES DO BRANCH SELF ON ERROR
001320  012737  000000  002130          MOV     #0,@#CPUHLT      ;IF NOT THEN PUT A HALT IN ON ERROR
001326  012737  000000  050520          MOV     #0,@#MMUHLT
001334  012737  000000  120472          MOV     #0,@#FPHLT
001342  012737  000000  124460          MOV     #0,@#LTCHLT
001350  012737  000000  123216          MOV     #0,@#SL1HLT
001356  012737  000000  127302          MOV     #0,@#SL2HLT
001364  012737  000000  130452          MOV     #0,@#COMHLT

                                ;*************************************************************************
                                ;
                                ;       THIS TEST EXECUTES EVERY POSSIBLE BRANCH WITH EVERY POSSIBLE
                                ;CONDITION CODE COMBINATION.
                                ;       THE ROUTINE USES TWO TABLES.  THE BRANCH TABLE HOLDS ALL THE
                                ;POSSIBLE BRANCH INSTRUCTIONS, THE OTHER TABLE (YNTAP) HOLDS BIT MAPS FOR
                                ;EACH BRANCH.  A ONE IN THE BIT MAP INDICATES THAT THE CORRESPONDING
                                ;BRANCH INSTRUCTION SHOULD BRANCH FOR THE CONDITION CODE SETTING WHICH
                                ;CORRESPONDS TO THE BIT POSITION WITHIN THE MAP.  FOR EXAMPLE IF THE LEFT
                                ;MOST BIT IS A ONE THEN THE CORRESPONDING BRANCH INSTRUCTION SHOULD BRANCH
                                ;WHEN THE CONDITION CODES ARE 0.
                                ;       THE ROUTINE CONSISTS OF NESTED LOOPS; THE OUTER LOOP SETS UP
                                ;ALL THE POSSIBLE BRANCH INSTRUCTIONS.  THE INNER LOOP SETS UP EVERY POSSIBLE
                                ;CONDITION CODE FOR EACH BRANCH.
                                ;       THE BIT MAP IS USED TO SET THE ADDRESS LOCATION IN TWO
                                ;JUMP MODE 3 INSTRUCTIONS.  THE ADDRESSES ARE CHANGED TO ALLOW THE
                                ;PROGRAM TO CONTINUE OR JUMP TO AN ERROR ROUTINE DEPENDING UPON
                                ;WHETHER IT HANDLED THE BRANCH INSTRUCTION CORRECTLY.
                                ;       AT ANY ERROR HALT, LOCATION, BRH, HOLDS THE BRANCH INSTRUCTION
                                ;UNDER TEST AND LOCATION, CC, HOLDS THE VALUE OF THE CONDITION CODES
                                ;AT THE TIME THE BRANCH WAS EXECUTED.
                                ;
                                ;*************************************************************************
                                ;TEST 1 TEST THE BRANCH ROM
                                ;*************************************************************************
001372                          TS1:
001372  012700  021162          SETUP:  MOV     #BRTAB,R0        ;INITIALIZE BRANCH TABLE POINTER
001376  012704  021220                  MOV     #YNTAB,R4        ;INITIALIZE YES/NO BRANCH MAP POINTER
001402  012767  000017  000130          MOV     #15.,BRCT        ;INITIALIZE BRANCH TABLE COUNT
001410  012067  000110          SETBR:  MOV     (R0)+,BRH        ;GET NEXT BRANCH INST.
001414  012401                          MOV     (R4)+,R1         ;GET NEXT BRANCH MAP
001416  012767  177777  000074          MOV     #-1,CC1          ;INITIALIZE CONDITION CODE VALUE
001424  012703  000020                  MOV     #16.,R3          ;INITIALIZE CONDITION CODE COUNT
001430  005267  000064          SETCC:  INC     CC1              ;SET FOR NEXT CC VALUE
001434  032701  100000                  BIT     #100000,R1       ;SEE IF SHOULD BR W/ THESE CC'S
001440  013705  177776                  MOV     @#177776,R5      ;SIMULATE A JNE
001444  042705  177773                  BIC     #177773,R5       ;       (JUMP NOT EQUAL)
001450  000165  001454                  JMP     .+4(R5)          ;       TO SET2BR
001454  000167  000020                  JMP     SET2BR
001460  012767  001542  000042          MOV     #CONT,NBR        ;SET TO CONTINUE IF NO BRANCH
001466  012767  001536  000040          MOV     #ER,YBR          ;SET TO REPORT ERROR IF BRANCH
001474  000167  000014                  JMP     AROUN            ;GO AROUND OPPOSITE CONDITION
001500  012767  001536  000022  SET2BR: MOV     #ER,NBR          ;SET TO REPORT ERROR IF NO BRANCH
001506  012767  001542  000020          MOV     #CONT,YBR        ;SET TO CONTINUE IF BRANCH
```

```
001514  006101                          AROUN:  ROL     R1                      ;UPDATE BIT MAP

001516  012737                                  MOV     (PC)+,@(PC)+            ;SET CONDITION CODE
001520  000000                          CC1:    0                               ;NEW CC VALUE GOES HERE
001522  177776                                  177776
001524  000000                          BRH:    0                               ;BRANCH INST. GOES HERE
001526  000137                                  JMP     @(PC)+                  ;THIS JUMP IF NO BRANCH
001530  000000                          NBR:    0                               ;WHERE TO GO IF NO BRANCH OCCURS
001532  000137                                  JMP     @(PC)+                  ;THIS JUMP IF BRANCH OCCURS
001534  000000                          YBR:    0                               ;WHERE TO GO IF BRANCH OCCURS
001536                                  ER:
001536  000551                                  BR      ERROR1                  ;
001540  000000                          BRCT:   0
001542  005303                          CONT:   DEC     R3                      ;CC'S DONE?
001544  013705  177776                          MOV     @#177776,R5             ;SIMULATE A JNE
001550  042705  177773                          BIC     #177773,R5              ;       (JUMP NOT EQUAL)
001554  000165  001560                          JMP     .+4(R5)                 ;       TO SETCC
001560  000167  177644                          JMP     SETCC
001564  005367  177750                          DEC     BRCT                    ;BR'S DONE?
001570  013705  177776                          MOV     @#177776,R5             ;SIMULATE A JNE
001574  042705  177773                          BIC     #177773,R5              ;       (JUMP NOT EQUAL)
001600  000165  001604                          JMP     .+4(R5)                 ;       TO SETBR
001604  000167  177600                          JMP     SETBR
001610  012700  000357                          MOV     #357,R0                 ;IF THIS TEST IS DONE SET UP R0 FOR THE NEXT
                                                                                ;SEVEN TESTS. THIS IS SAVING 4 LOCATIONS PER
                                                                                ;TEST WHICH I NEED BECAUSE BRANCHES WERE OUT
                                                                                ;OF BOUNDS.



                                        ;*************************************************************************
                                        ;TEST 2 TEST TRAP OF RESERVED INSTRUCTION
                                        ;*************************************************************************
001614                                  TS2:
001614  012706  001000                          MOV     #STBOT,SP               ;INITIALIZE THE STACK POINTER
001620  012767  001642  176162                  MOV     #1$,RTRAP                       ;SET UP NEW PC IN VECTOR
001626  010067  176160                          MOV     R0,RTRAP+2                      ;SET UP NEW PSW IN VECTOR
001632  005067  176140                          CLR     STATUS                  ;CLEAR PRESENT (OLD) STATUS
001636  000077                                  TRAPA                           ;DO TRAP
001640                                  8$:
001640  000510                                  BR      ERROR1                  ;INSTRUCTION FAILED TO TRAP
001642  020067  176130                  1$:     CMP     R0,STATUS               ;IS NEW STATUS CORRECT
001646  001105                                  BNE     ERROR1                  ;NEW STATUS WRONG
001650  020627  000774                  2$:     CMP     SP,#STBOT-4             ;DID STACK DECREMENT CORRECTLY
001654  001102                                  BNE     ERROR1                  ;STACK DID NOT DECREMENT CORRECTLY
001656  022767  001640  177110          3$:     CMP     #8$,STBOT-4             ;WAS PROPER PC SAVED
001664  001076                                  BNE     ERROR1                  ;PROPER PC WAS NOT SAVED
001666  005767  177104                  4$:     TST     STBOT-2                 ;WAS OLD PSW SAVED
001672  001073                                  BNE     ERROR1                  ;WRONG PSW SAVED
001674  062706  000004                  5$:     ADD     #4,SP                   ;RESET STACK POINTER
001700  012767  001722  176102                  MOV     #6$,RTRAP                       ;SET UP NEW PC IN VECTOR
001706  005067  176100                          CLR     RTRAP+2                 ;SET UP NEW PSW IN VECTOR
001712  010037  177776                          MOV     R0,@#STATUS             ;SET UP OLD STATUS FOR COMPARISON AFTER TRAP
001716  000077                                  TRAPA                           ;DO TRAP
001720  000460                                  BR      ERROR1                  ;INSTRUCTION FAILED TO TRAP
001722  005767  176050                  6$:     TST     STATUS                  ;IS NEW PSW CORRECT
001726  001055                                  BNE     ERROR1                  ;NEW PSW WRONG
```

```
001730 020067 177042          7$:     CMP     R0,STBOT-2      ;WAS OLD STATUS STORED
001734 001052                         BNE     ERROR1          ;OLD STATUS WRONG
                              ;*****************************************************************************
                              ;TEST 3 TEST TRAP OF TRAP INSTRUCTION
                              ;*****************************************************************************
001736                        TS3:
001736 012706 001000                  MOV     #STBOT,SP       ;INITIALIZE THE STACK POINTER
001742 012767 001764 176064           MOV     #1$,RTRAP1              ;SET UP NEW PC IN VECTOR
001750 010067 176062                  MOV     R0,RTRAP1+2             ;SET UP NEW PSW IN VECTOR
001754 005067 176016                  CLR     STATUS          ;CLEAR PRESENT (OLD) STATUS
001760 104400                         TRAP                    ;DO TRAP
001762                        8$:
001762 000437                         BR      ERROR1          ;INSTRUCTION FAILED TO TRAP
001764 020067 176006          1$:     CMP     R0,STATUS       ;IS NEW STATUS CORRECT
001770 001034                         BNE     ERROR1          ;NEW STATUS WRONG
001772 020627 000774          2$:     CMP     SP,#STBOT-4     ;DID STACK DECREMENT CORRECTLY
001776 001031                         BNE     ERROR1          ;STACK DID NOT DECREMENT CORRECTLY
002000 022767 001762 176766   3$:     CMP     #8$,STBOT-4     ;WAS PROPER PC SAVED
002006 001025                         BNE     ERROR1          ;PROPER PC WAS NOT SAVED
002010 005767 176762          4$:     TST     STBOT-2         ;WAS OLD PSW SAVED
002014 001022                         BNE     ERROR1          ;WRONG PSW SAVED
002016 062706 000004          5$:     ADD     #4,SP           ;RESET STACK POINTER
002022 012767 002044 176004           MOV     #6$,RTRAP1              ;SET UP NEW PC IN VECTOR
002030 005067 176002                  CLR     RTRAP1+2                ;SET UP NEW PSW IN VECTOR
002034 010037 177776                  MOV     R0,@#STATUS     ;SET UP OLD STATUS FOR COMPARISON AFTER TRAP
002040 104777                         TRAPC                   ;DO TRAP WITH LOWER BYTE ALL ONES
002042 000407                         BR      ERROR1          ;INSTRUCTION FAILED TO TRAP
002044 005767 175726          6$:     TST     STATUS          ;IS NEW PSW CORRECT
002050 001004                         BNE     ERROR1          ;NEW PSW WRONG
002052 020067 176720          7$:     CMP     R0,STBOT-2      ;WAS OLD STATUS STORED
002056 001001                         BNE     ERROR1          ;OLD STATUS WRONG
002060 000424                         BR      CPUHLT+2        ;GET OVER ERROR CALL TO NEXT TEST IF NO ERROR
                              ;*****************************************************************************
                              ;THIS ERROR IS USED FOR THE ENTIRE CPU,TRAPS AND EIS PORTION OF
                              ;THIS TEST
                              ;*****************************************************************************
002062 012737 002114 000004   ERROR1: MOV     #1$,@#4         ;SET UP THE TIMEOUT VECTOR
002070 012737 000340 000006           MOV     #340,@#6
002076 005737 177766                  TST     @#177766        ;CHECK TO SEE IF THE POWER FAILED,WILL TRAP
                                                              ;TO 1$ IF UNIBUS MAP BOARD IS NOT PRESENT
                                                              ;SINCE THE CPUERR(177766)IS ON THE UNIBUS
                                                              ;MAP BOARD
002102 001404                         BEQ     1$              ;NO,GO REPORT THE ERROR
002104 012737 000177 001002           MOV     #177,@#$FATAL   ;YES,FLAG THAT THE POWER FAILED
002112 000403                         BR      2$              ; AND GO SET FATAL ERROR FLAG
002114 012737 000001 001002   1$:     MOV     #1,@#$FATAL                     ;SET UP FATAL ERROR NUMBER
002122 012767 000001 176650   2$:     MOV     #1,$MSGTY                       ;SET FATAL ERROR FLAG
002130 000777                 CPUHLT: BR      .
                              ;*****************************************************************************
                              ;TEST 4 TEST TRAP OF IOT INSTRUCTION
                              ;*****************************************************************************
002132                        TS4:
002132 012706 001000                  MOV     #STBOT,SP       ;INITIALIZE THE STACK POINTER
002136 012767 002160 175654           MOV     #1$,RTRAP2              ;SET UP NEW PC IN VECTOR
002144 010067 175652                  MOV     R0,RTRAP2+2             ;SET UP NEW PSW IN VECTOR
002150 005067 175622                  CLR     STATUS          ;CLEAR PRESENT (OLD) STATUS
```

```
002154  000004                          IOT                     ;DO TRAP
002156                          8$:
002156  000741                          BR      ERROR1          ;INSTRUCTION FAILED TO TRAP
002160  020067  175612          1$:     CMP     R0,STATUS       ;IS NEW STATUS CORRECT
002164  001336                          BNE     ERROR1          ;NEW STATUS WRONG
002166  020627  000774          2$:     CMP     SP,#STBOT-4     ;DID STACK DECREMENT CORRECTLY
002172  001333                          BNE     ERROR1          ;STACK DID NOT DECREMENT CORRECTLY
002174  022767  002156  176572  3$:     CMP     #8$,STBOT-4     ;WAS PROPER PC SAVED
002202  001327                          BNE     ERROR1          ;PROPER PC WAS NOT SAVED
002204  005767  176566          4$:     TST     STBOT-2         ;WAS OLD PSW SAVED
002210  001324                          BNE     ERROR1          ;WRONG PSW SAVED
002212  062706  000004          5$:     ADD     #4,SP           ;RESET STACK POINTER
002216  012767  002240  175574          MOV     #6$,RTRAP2              ;SET UP NEW PC IN VECTOR
002224  005067  175572                  CLR     RTRAP2+2                ;SET UP NEW PSW IN VECTOR
002230  010037  177776                  MOV     R0,@#STATUS     ;SET UP OLD STATUS FOR COMPARISON AFTER TRAP
002234  000004                          IOT                     ;DO TRAP
002236  000711                          BR      ERROR1          ;INSTRUCTION FAILED TO TRAP
002240  005767  175532          6$:     TST     STATUS          ;IS NEW PSW CORRECT
002244  001306                          BNE     ERROR1          ;NEW PSW WRONG
002246  020067  176524          7$:     CMP     R0,STBOT-2      ;WAS OLD STATUS STORED
002252  001303                          BNE     ERROR1          ;OLD STATUS WRONG
                                ;*******************************************************************************
                                ;TEST 5 TEST TRAP OF EMT INSTRUCTION
                                ;*******************************************************************************
002254                          TS5:
002254  012706  001000                  MOV     #STBOT,SP       ;INITIALIZE THE STACK POINTER
002260  012767  002302  175542          MOV     #1$,RTRAP3              ;SET UP NEW PC IN VECTOR
002266  010067  175540                  MOV     R0,RTRAP3+2             ;SET UP NEW PSW IN VECTOR
002272  005067  175500                  CLR     STATUS          ;CLEAR PRESENT (OLD) STATUS
002276  104000                          EMT                     ;DO TRAP
002300                          8$:
002300  000670                          BR      ERROR1          ;INSTRUCTION FAILED TO TRAP
002302  020067  175470          1$:     CMP     R0,STATUS       ;IS NEW STATUS CORRECT
002306  001265                          BNE     ERROR1          ;NEW STATUS WRONG
002310  020627  000774          2$:     CMP     SP,#STBOT-4     ;DID STACK DECREMENT CORRECTLY
002314  001262                          BNE     ERROR1          ;STACK DID NOT DECREMENT CORRECTLY
002316  022767  002300  176450  3$:     CMP     #8$,STBOT-4     ;WAS PROPER PC SAVED
002324  001256                          BNE     ERROR1          ;PROPER PC WAS NOT SAVED
002326  005767  176444          4$:     TST     STBOT-2         ;WAS OLD PSW SAVED
002332  001253                          BNE     ERROR1          ;WRONG PSW SAVED
002334  062706  000004          5$:     ADD     #4,SP           ;RESET STACK POINTER
002340  012767  002362  175462          MOV     #6$,RTRAP3              ;SET UP NEW PC IN VECTOR
002346  005067  175460                  CLR     RTRAP3+2                ;SET UP NEW PSW IN VECTOR
002352  010037  177776                  MOV     R0,@#STATUS     ;SET UP OLD STATUS FOR COMPARISON AFTER TRAP
002356  104377                          EMTA                    ;DO TRAP WITH LOWER BYTE ALL ONES
002360  000640                          BR      ERROR1          ;INSTRUCTION FAILED TO TRAP
002362  005767  175410          6$:     TST     STATUS          ;IS NEW PSW CORRECT
002366  001235                          BNE     ERROR1          ;NEW PSW WRONG
002370  020067  176402          7$:     CMP     R0,STBOT-2      ;WAS OLD STATUS STORED
002374  001232                          BNE     ERROR1          ;OLD STATUS WRONG
002376  000401                          BR      ERROR2+2        ;WE MUST GET OVER ERROR CALL AT END OF THIS TEST
                                ;*******************************************************************************
                                ;THIS ERROR IS NEEDED BECAUSE BRANCHES IN TRAP TESTS BEYOND HERE CAN NOT
                                ;REACH ERROR1.
                                ;*******************************************************************************
002400  000630                  ERROR2: BR      ERROR1
```

```


                                            ;*****************************************************************************
                                            ;TEST 6 TEST TRAP OF TRACE-TRAP INSTRUCTION
                                            ;*****************************************************************************
 002402                                     TS6:
 002402  012706  001000                             MOV     #STBOT,SP       ;INITIALIZE THE STACK POINTER
 002406  012767  002430  175400                     MOV     #1$,RTRAP4              ;SET UP NEW PC IN VECTOR
 002414  010067  175376                             MOV     R0,RTRAP4+2             ;SET UP NEW PSW IN VECTOR
 002420  005067  175352                             CLR     STATUS          ;CLEAR PRESENT (OLD) STATUS
 002424  000003                                     TRT                     ;DO TRAP
 002426                                     8$:
 002426  000764                                     BR      ERROR2          ;INSTRUCTION FAILED TO TRAP
 002430  020067  175342                     1$:     CMP     R0,STATUS       ;IS NEW STATUS CORRECT
 002434  001361                                     BNE     ERROR2          ;NEW STATUS WRONG
 002436  020627  000774                     2$:     CMP     SP,#STBOT-4     ;DID STACK DECREMENT CORRECTLY
 002442  001356                                     BNE     ERROR2          ;STACK DID NOT DECREMENT CORRECTLY
 002444  022767  002426  176322             3$:     CMP     #8$,STBOT-4     ;WAS PROPER PC SAVED
 002452  001352                                     BNE     ERROR2          ;PROPER PC WAS NOT SAVED
 002454  005767  176316                     4$:     TST     STBOT-2         ;WAS OLD PSW SAVED
 002460  001347                                     BNE     ERROR2          ;WRONG PSW SAVED
 002462  062706  000004                     5$:     ADD     #4,SP           ;RESET STACK POINTER
 002466  012767  002510  175320                     MOV     #6$,RTRAP4              ;SET UP NEW PC IN VECTOR
 002474  005067  175316                             CLR     RTRAP4+2                ;SET UP NEW PSW IN VECTOR
 002500  010037  177776                             MOV     R0,@#STATUS     ;SET UP OLD STATUS FOR COMPARISON AFTER TRAP
 002504  000003                                     TRT                     ;DO TRAP
 002506  000734                                     BR      ERROR2          ;INSTRUCTION FAILED TO TRAP
 002510  005767  175262                     6$:     TST     STATUS          ;IS NEW PSW CORRECT
 002514  001331                                     BNE     ERROR2          ;NEW PSW WRONG
 002516  020067  176254                     7$:     CMP     R0,STBOT-2      ;WAS OLD STATUS STORED
 002522  001326                                     BNE     ERROR2          ;OLD STATUS WRONG
                                            ;PDP-11 ILLEGAL AND ADDRESS INSTRUCTION TEST
                                            ;ALL INSTRUCTIONS THAT ARE RESERVED
                                            ;SHOULD TRAP TO LOCATION 4, AND THE
                                            ;PC THAT POINTS TO THE TRAPPING INSTRUCTION
                                            ;SHOULD BE PLACED ON THE STACK

                                            ;*****************************************************************************
                                            ;TEST 7 TEST TRAP OF ILLEGAL INSTRUCTION
                                            ;*****************************************************************************
 002524                                     TS7:
 002524  012706  001000                             MOV     #STBOT,SP       ;INITIALIZE THE STACK POINTER
 002530  012767  002552  175246                     MOV     #1$,RTRAP5              ;SET UP NEW PC IN VECTOR
 002536  010067  175244                             MOV     R0,RTRAP5+2             ;SET UP NEW PSW IN VECTOR
 002542  005067  175230                             CLR     STATUS          ;CLEAR PRESENT (OLD) STATUS
 002546  000100                                     JMP     %0                      ;DO TRAP
 002550                                     8$:
 002550  000713                                     BR      ERROR2          ;INSTRUCTION FAILED TO TRAP
 002552  020067  175220                     1$:     CMP     R0,STATUS       ;IS NEW STATUS CORRECT
 002556  001310                                     BNE     ERROR2          ;NEW STATUS WRONG
 002560  020627  000774                     2$:     CMP     SP,#STBOT-4     ;DID STACK DECREMENT CORRECTLY
 002564  001305                                     BNE     ERROR2          ;STACK DID NOT DECREMENT CORRECTLY
 002566  022767  002550  176200             3$:     CMP     #8$,STBOT-4     ;WAS PROPER PC SAVED
 002574  001301                                     BNE     ERROR2          ;PROPER PC WAS NOT SAVED
 002576  005767  176174                     4$:     TST     STBOT-2         ;WAS OLD PSW SAVED
 002602  001276                                     BNE     ERROR2          ;WRONG PSW SAVED
```

```
002604 062706 000004          5$:   ADD   #4,SP           ;RESET STACK POINTER
002610 012767 002632 175166         MOV   #6$,RTRAP5              ;SET UP NEW PC IN VECTOR
002616 005067 175164                CLR   RTRAP5+2                ;SET UP NEW PSW IN VECTOR
002622 010037 177776                MOV   R0,@#STATUS     ;SET UP OLD STATUS FOR COMPARISON AFTER TRAP
002626 000100                       JMP   %0                      ;DO TRAP
002630 000663                       BR    ERROR2          ;INSTRUCTION FAILED TO TRAP
002632 005767 175140          6$:   TST   STATUS          ;IS NEW PSW CORRECT
002636 001260                       BNE   ERROR2          ;NEW PSW WRONG
002640 020067 176132          7$:   CMP   R0,STBOT-2      ;WAS OLD STATUS STORED
002644 001255                       BNE   ERROR2          ;OLD STATUS WRONG
                              ;*************************************************************************
                              ;TEST 10      TEST TRAP OF ALL ILLEGAL INSTRUCTION
                              ;*************************************************************************
002646                        TS10:
002646 012706 001000                MOV   #STBOT,SP       ;INITIALIZE THE STACK POINTER
002652 012767 002674 175124         MOV   #1$,RTRAP5              ;SET UP NEW PC IN VECTOR
002660 010067 175122                MOV   R0,RTRAP5+2             ;SET UP NEW PSW IN VECTOR
002664 005067 175106                CLR   STATUS          ;CLEAR PRESENT (OLD) STATUS
002670 004000                       JSR   %0,%0                   ;DO TRAP
002672                        8$:
002672 000642                       BR    ERROR2          ;INSTRUCTION FAILED TO TRAP
002674 020067 175076          1$:   CMP   R0,STATUS       ;IS NEW STATUS CORRECT
002700 001237                       BNE   ERROR2          ;NEW STATUS WRONG
002702 020627 000774          2$:   CMP   SP,#STBOT-4     ;DID STACK DECREMENT CORRECTLY
002706 001234                       BNE   ERROR2          ;STACK DID NOT DECREMENT CORRECTLY
002710 022767 002672 176056   3$:   CMP   #8$,STBOT-4     ;WAS PROPER PC SAVED
002716 001230                       BNE   ERROR2          ;PROPER PC WAS NOT SAVED
002720 005767 176052          4$:   TST   STBOT-2         ;WAS OLD PSW SAVED
002724 001225                       BNE   ERROR2          ;WRONG PSW SAVED
002726 062706 000004          5$:   ADD   #4,SP           ;RESET STACK POINTER
002732 012767 002754 175044         MOV   #6$,RTRAP5              ;SET UP NEW PC IN VECTOR
002740 005067 175042                CLR   RTRAP5+2                ;SET UP NEW PSW IN VECTOR
002744 010037 177776                MOV   R0,@#STATUS     ;SET UP OLD STATUS FOR COMPARISON AFTER TRAP
002750 004000                       JSR   %0,%0                   ;DO TRAP
002752 000612                       BR    ERROR2          ;INSTRUCTION FAILED TO TRAP
002754 005767 175015          6$:   TST   STATUS          ;IS NEW PSW CORRECT
002760 001207                       BNE   ERROR2          ;NEW PSW WRONG
002762 020067 176010          7$:   CMP   R0,STBOT-2      ;WAS OLD STATUS STORED
002766 001204                       BNE   ERROR2          ;OLD STATUS WRONG

                              ;*************************************************************************
                              .SBTTL  DATA PATH TESTS
                              ;
                              ;       THE DATA PATH TESTS ARE USED TO VERIFY THAT VARIOUS
                              ;DATA PATTERNS CAN BE SUCCESSFULLY MOVED THROUGH THE DATA PATHS
                              ;MOVE AND COMPARE MODE 2,3 INSTRUCTIONS ARE USED TO PASS AND
                              ;TEST VARIOUS DATA PATTERNS IN THE DATA PATHS.
                              ;       THE TEST EXERCISES THE INTERNAL DATA PATHS,  AND THE UNIBUS
                              ;DATA TRANSCIEVERS.
                              ;       IF THESE TESTS FAIL, EXAMINE THE TARGET LOCATION (LOC. 0)
                              ;TO SEE WHICH BITS OF THE DATA PATH ARE FAILING.
002770 012737 002062 000030         MOV   #ERROR1,@#30    ;SET UP VECTOR FOR ERROR CALLS
002776 012737 000340 000032         MOV   #340,@#32       ;SET UP NEW PSW
003004 012737 021256 000004         MOV   #T04,@#4        ;SET UP FOR UNEXPECTED TRAP TO 4
003012 012737 021260 000010         MOV   #T010,@#10      ;SET UP FOR UNEXPECTED TRAP TO 10
003020 012737 021262 000014         MOV   #T014,@#14      ;SET UP FOR UNEXPECTED TRAP TO 14
```

```
003026  012737  021270  000034          MOV     #T034,@#34          ;SET UP FOR UNEXPECTED TRAP TO 34
003034  012737  021264  000020          MOV     #T020,@#20          ;SET UP FOR UNEXPECTED TRAP TO 20


                                ;*******************************************************************************
                                ;TEST 11          TEST OF ZEROES IN THE DATA PATH
                                ;*******************************************************************************
003042                          TS11:
003042  012737  000000  000000          MOV     #0,@#0              ;MOVE ZEROES THRU ADDRESS LINES, DATA
                                                                    ;LINES AND INTERNAL PATHS
003050  005737  000000                  TST     @#0                 ;SUCCESSFUL?
003054  001401                          BEQ     TS12
003056  104000                          EMT                         ;DATA INCORRECT

                                ;*******************************************************************************
                                ;TEST 12          TEST OF PATTERN 125252 IN DATA PATH
                                ;*******************************************************************************
003060                          TS12:
003060  012737  125252  000000          MOV     #125252,@#0         ;MOVE ALTERNATING ONES AND ZEROES
                                                                    ;THRU DATA PATHS
003066  022737  125252  000000          CMP     #125252,@#0         ;SUCCESSFUL
003074  001401                          BEQ     TS13
003076  104000                          EMT                         ;DATA INCORRECT

                                ;*******************************************************************************
                                ;TEST 13          TEST OF PATTERN 052525 IN DATA PATH
                                ;*******************************************************************************
003100                          TS13:
003100  012737  052525  000000          MOV     #052525,@#0         ;MOVE ALTERNATING ZEROES AND ONES
                                                                    ;THRU DATA PATH
003106  022737  052525  000000          CMP     #052525,@#0         ;SUCCESSFUL?
003114  001401                          BEQ     TS14
003116  104000                          EMT                         ;DATA INCORRECT

                                ;*******************************************************************************
                                ;TEST 14          TEST OF ALL ONES IN DATA PATH
                                ;*******************************************************************************
003120                          TS14:
003120  012737  177777  000000          MOV     #177777,@#0         ;MOVE ONES THRU DATA PATH
003126  022737  177777  000000          CMP     #177777,@#0         ;SUCCESSFUL
003134  001401                          BEQ     TS15
003136  104000                          EMT                         ;DATA INCORRECT

                                ;*******************************************************************************
                                .SBTTL  B-REGISTER TEST
                                ;
                                ;       THE B-REGISTER (LOCATION 0) SHIFTING LOGIC TESTS ARE USED
                                ;TO TEST THAT THE B-REGISTER CAN HOLD VARIOUS DATA PATTERNS AND THAT
                                ;THE ASSOCIATED LOGIC SUPPORTS THE SHIFTING FUNCTIONS WITHIN THE
                                ;B-REGISTER AND C-BIT.
                                ;A ONE IS SHIFTED THROUGH EVERY BIT IN THE B-REGISTER AND C-BIT IN
                                ;BOTH DIRECTIONS.
                                ;       THE B-REGISTER ITSELF IS TESTED IN ITS ABILITY AS A BUFFER AND AS
                                ;A SHIFT REGISTER.  DATA IS ALSO PASSED THROUGH THE DATA PATH AND ALU,
                                ;       IF THESE TESTS FAIL, EXAMINE THE TARGET LOCATION (LOC. 0) TO SEE
                                ;WHICH BITS OF THE B-REGISTER MAY BE FAILING.
```

```
                                        ;
                                        ;*******************************************************************
                                        ;TEST 15        SHIFT BIT 0 TO BIT 1
                                        ;*******************************************************************
003140                                  TS15:
003140  000241                                  CLC                             ;CLEAR CARRY BIT
003142  012737  000001  000000                  MOV     #1,@#0                  ;LOAD A 1
003150  006137  000000                          ROL     @#0                     ;SHIFT LEFT
003154  022737  000002  000000                  CMP     #2,@#0                  ;SUCCESSFUL
003162  001401                                  BEQ     TS16
003164  104000                                  EMT                             ;BIT 1 NOT SET

                                        ;*******************************************************************
                                        ;TEST 16        SHIFT CARRY INTO BIT 0
                                        ;*******************************************************************
003166                                  TS16:
003166  012737  000000  000000                  MOV     #0,@#0                  ;CLEAR LOCATION
003174  000261                                  SEC                             ;SET CARRY
003176  006137  000000                          ROL     @#0                     ;ROTATE CARRY BIT TO BIT 0
003202  103001                                  BCC     CARRY1
003204  104000                                  EMT                             ;CARRY CLEAR
003206  022737  000001  000000          CARRY1: CMP     #1,@#0                  ;BIT 0 SET
003214  001401                                  BEQ     TS17
003216  104000                                  EMT                             ;BIT 0 NOT SET

                                        ;*******************************************************************
                                        ;TEST 17        LEFT SHIFT FROM BIT 0 TO C-BIT
                                        ;*******************************************************************
003220                                  TS17:
003220  012737  000001  000000                  MOV     #1,@#0                  ;SET BIT 0
003226  012700  177757                          MOV     #-21,R0                 ;SET BIT COUNTER
003232  000241                                  CLC                             ;CLEAR C-BIT
003234  005200                          SHL:    INC     R0                      ;INCREMENT BIT COUNTER
003236  001404                                  BEQ     SHLE                    ;BR TO ERROR HALT IF BIT IS LOST
003240  006137  000000                          ROL     @#0                     ;SHIFT LEFT ONE POSITION
003244  103373                                  BCC     SHL                     ;BRANCH IF C-BIT NOT SET
003246  001401                                  BEQ     TS20
003250                                  SHLE:
003250  104000                                  EMT                             ;LEFT SHIFTING LOGIC FAILED

                                        ;*******************************************************************
                                        ;TEST 20        SHIFT BIT 15 TO BIT 14
                                        ;*******************************************************************
003252                                  TS20:
003252  012737  100000  000000                  MOV     #100000,@#0             ;SET BIT 15
003260  000241                                  CLC                             ;CLEAR CARRY
003262  006037  000000                          ROR     @#0                     ;SHIFT BIT 15 TO BIT 14
003266  022737  040000  000000                  CMP     #40000,@#0              ;SUCCESSFUL
003274  001401                                  BEQ     TS21
003276  104000                                  EMT                             ;BIT 14 NOT SET

                                        ;*******************************************************************
                                        ;TEST 21        RIGHT SHIFT FROM BIT 15 TO C-BIT
                                        ;*******************************************************************
003300                                  TS21:
003300  012737  100000  000000                  MOV     #100000,@#0             ;SET BIT 15
```

```
003306  012700  177757                  MOV     #-21,R0          ;SET BIT COUNTER
003312  000241                          CLC                      ;CLEAR C-BIT
003314  005200                  SHR:    INC     R0               ;INCREMENT BIT COUNTER
003316  001404                          BEQ     SHRE             ;BR TO ERROR HALT IF BIT IS LOST
003320  006037  000000                  ROR     @#0              ;ROTATE RIGHT ONE POSITION
003324  103373                          BCC     SHR              ;BRANCH IF C-BIT CLEAR
003326  001401                          BEQ     TS22
003330                          SHRE:
003330  104000                          EMT                      ;RIGHT SHIFT LOGIC FAILED

                                ;*******************************************************************************
                                .SBTTL  SCRATCH PAD TESTS
                                ;
                                ;       THE SCRATCH PAD TESTS ARE USED TO VERIFY THAT VARIOUS
                                ;DATA PATTERNS CAN BE SUCCESSFULLY HELD IN THE SCRATCH PAD
                                ;CIRCUITRY.  MOVE AND COMPARE INSTRUCTIONS ARE USED TO TEST THAT
                                ;R0 CAN HOLD VARIOUS DATA PATTERNS.  EACH DATA PATTERN IS
                                ;MOVED AND TESTED IN A SMALL LOOP CONVENIENT FOR SCOPING.  THE
                                ;SUCCESSFUL COMPLETION OF THESE TESTS SHOULD VERIFY THE CIRCUITRY EXTERNAL
                                ;TO THE SCRATCH PAD ITSELF.
                                ;       THE REMAINDER OF THE GENERAL REGISTERS ARE TESTED BY MOVING
                                ;A BIT INTO BIT 0 OF THE REGISTER AND SHIFTING IT LEFT ONE
                                ;BIT AT A TIME INTO THE CARRY BIT.  THE RESULT IS THEN CHECKED TO INSURE THAT
                                ;NO BITS WERE PICKED. THE PROCEDURE IS THEN REPEATED UNDER OPPOSITE
                                ;CONDITIONS. THE GENERAL REGISTER AND THE CARRY BIT ARE SET TO
                                ;ALL ONES, AND A ZERO IS SHIFTED LEFT FROM BIT 0 INTO THE CARRY BIT.
                                ;THE RESULT IS THEN CHECKED TO INSURE THAT NO ZEROES WERE PICKED.
                                ;       AT THIS POINT ALL OF THE GENERAL REGISTERS HAVE BEEN EXERCISED
                                ;AS WELL AS REGISTER 11.
                                ;
                                ;*******************************************************************************
                                ;TEST 22        TEST IF R0 CAN HOLD ALL ZEROES
                                ;*******************************************************************************
003332                          TS22:

003332  012700  000000                  MOV     #0,R0            ;MOVE ZEROES TO R0
003336  005700                          TST     R0               ;SUCCESSFUL?
003340  001401                          BEQ     TS23
003342  104000                          EMT                      ;R0 NOT 0

                                ;*******************************************************************************
                                ;TEST 23        TEST IF R0 CAN HOLD ONES AND ZEROES
                                ;*******************************************************************************
003344                          TS23:
003344  012700  125252                  MOV     #125252,R0       ;MOVE ALTERNATING ONES AND ZEROES TO R0
003350  020027  125252                  CMP     R0,#125252       ;SUCCESSFUL?
003354  001401                          BEQ     TS24
003356  104000                          EMT                      ;R0 NOT 125252

                                ;*******************************************************************************
                                ;TEST 24        TEST IF R0 CAN HOLD ZEROES AND ONES
                                ;*******************************************************************************
003360                          TS24:
003360  012700  052525                  MOV     #052525,R0       ;MOVE ALTERNATING ZEROES AND ONES TO R0
003364  020027  052525                  CMP     R0,#052525       ;SUCCESSFUL?
003370  001401                          BEQ     TS25
```

```
003372  104000                      EMT                     ;R0 NOT 52525

                            ;***********************************************************************
                            ;TEST 25        TEST IF R0 CAN HOLD ALL ONES
                            ;***********************************************************************
003374                      TS25:
003374  012700  177777              MOV     #177777,R0      ;MOVE ALL ONES TO R0
003400  020027  177777              CMP     R0,#177777      ;SUCCESSFUL?
003404  001401                      BEQ     TS26
003406  104000                      EMT                     ;R0 NOT 177777

                            ;***********************************************************************
                            ;TEST 26        TEST IF R1 CAN HOLD A ONE IN ALL BITS
                            ;***********************************************************************
003410                      TS26:
003410  012701  000001              MOV     #1,R1           ;SET BIT 0
003414  012700  177757              MOV     #-21,R0         ;SET BIT COUNTER
003420  000241                      CLC                     ;CLEAR C-BIT
003422  005200              REG1:   INC     R0              ;INCREMENT BIT COUNTER
003424  001403                      BEQ     REG1E           ;BR TO ERROR HALT IF BIT IS LOST
003426  006101                      ROL     R1              ;ROTATE 1 POSITION
003430  103374                      BCC     REG1            ;ALL DONE
003432  001401                      BEQ     TS27
003434                      REG1E:
003434  104000                      EMT                     ;FAILURE WITH R1

                            ;***********************************************************************
                            ;TEST 27        TEST IF R1 CAN HOLD A ZERO IN ALL BITS
                            ;***********************************************************************
003436                      TS27:
003436  012701  177776              MOV     #-2,R1          ;SET ALL ONES IN R1 EXCEPT FOR BIT 0
003442  012700  177757              MOV     #-21,R0         ;SET BIT COUNTER
003446  000261                      SEC                     ;SET C-BIT
003450  005200              REG1A:  INC     R0              ;INCREMENT COUNTER
003452  001405                      BEQ     R1ERR           ;BR TO ERROR HALT IF COUNTER=0
003454  006101                      ROL     R1              ;ROTATE 1 POSITION
003456  103774                      BCS     REG1A           ;CONTINUE UNTIL C-BIT IS CLEAR
003460  022701  177777              CMP     #-1,R1          ;CHECK DATA IN R1
003464  001401                      BEQ     TS30
003466                      R1ERR:
003466  104000                      EMT                     ;FAILURE WITH R1
                            ;***********************************************************************
                            ;TEST 30        TEST IF R2 CAN HOLD A ONE IN ALL BITS
                            ;***********************************************************************
003470                      TS30:
003470  012702  000001              MOV     #1,R2           ;SET BIT 0
003474  012700  177757              MOV     #-21,R0         ;SET BIT COUNTER
003500  000241                      CLC                     ;CLEAR C-BIT
003502  005200              REG2:   INC     R0              ;INCREMENT BIT COUNTER
003504  001403                      BEQ     REG2E           ;BR TO ERROR HALT IF BIT IS LOST
003506  006102                      ROL     R2              ;ROTATE 1 POSITION
003510  103374                      BCC     REG2            ;ALL DONE
003512  001401                      BEQ     TS31
003514                      REG2E:
003514  104000                      EMT                     ;FAILURE WITH R2
```

```
                                ;*****************************************************************************
                                ;TEST 31        TEST IF R2 CAN HOLD A ZERO IN ALL BITS
                                ;*****************************************************************************
003516                          TS31:
003516  012702  177776                  MOV     #-2,R2          ;SET ALL ONES IN R2 EXCEPT FOR BIT 0
003522  012700  177757                  MOV     #-21,R0         ;SET BIT COUNTER
003526  000261                          SEC                     ;SET C-BIT
003530  005200                  REG2B:  INC     R0              ;INCREMENT BIT COUNTER
003532  001405                          BEQ     R2ERR           ;BR TO ERROR HALT IF COUNTER=0
003534  006102                          ROL     R2              ;ROTATE 1 POSITION
003536  103774                          BCS     REG2B           ;CONTINUE UNTIL C-BIT IS CLEAR
003540  022702  177777                  CMP     #-1,R2          ;CHECK DATA IN R2
003544  001401                          BEQ     TS32
003546                          R2ERR:
003546  104000                          EMT                     ;FAILURE WITH R2

                                ;*****************************************************************************
                                ;TEST 32        TEST IF R3 CAN HOLD A ONE IN ALL BITS
                                ;*****************************************************************************
003550                          TS32:
003550  012703  000001                  MOV     #1,R3           ;SET BIT 0
003554  012700  177757                  MOV     #-21,R0         ;SET BIT COUNTER
003560  000241                          CLC                     ;CLEAR C-BIT
003562  005200                  REG3:   INC     R0              ;INCREMENT BIT COUNTER
003564  001403                          BEQ     REG3E           ;BR TO ERROR HALT IF BIT IS LOST
003566  006103                          ROL     R3              ;ROTATE 1 POSITION
003570  103374                          BCC     REG3            ;ALL DONE
003572  001401                          BEQ     TS33
003574                          REG3E:
003574  104000                          EMT                     ;FAILURE WITH R3

                                ;*****************************************************************************
                                ;TEST 33        TEST IF R3 CAN HOLD A ZERO IN ALL BITS
                                ;*****************************************************************************
003576                          TS33:
003576  012703  177776                  MOV     #-2,R3          ;SET ALL ONES IN R3 EXCEPT FOR BIT 0
003602  012700  177757                  MOV     #-21,R0         ;SET BIT COUNTER
003606  000261                          SEC                     ;SET C-BIT
003610  005200                  REG3A:  INC     R0              ;INCREMENT BIT COUNTER
003612  001405                          BEQ     R3ERR           ;BR TO ERROR HALT IF COUNTER=0
003614  006103                          ROL     R3              ;ROTATE 1 POSITION
003616  103774                          BCS     REG3A           ;CONTINUE UNTIL C-BIT IS CLEAR
003620  022703  177777                  CMP     #-1,R3          ;CHECK DATA
003624  001401                          BEQ     TS34
003626                          R3ERR:
003626  104000                          EMT                     ;FAILURE WITH R3

                                ;*****************************************************************************
                                ;TEST 34        TEST IF R4 CAN HOLD A ONE IN ALL BITS
                                ;*****************************************************************************
003630                          TS34:
003630  012704  000001                  MOV     #1,R4           ;SET BIT 0
003634  012700  177757                  MOV     #-21,R0         ;SET BIT COUNTER
003640  000241                          CLC                     ;CLEAR C-BIT
003642  005200                  REG4:   INC     R0              ;INCREMENT BIT COUNTER
003644  001403                          BEQ     REG4E           ;BR TO ERROR HALT IF BIT IS LOST
```

```
003646  006104                          ROL     R4              ;ROTATE 1 POSITION
003650  103374                          BCC     REG4            ;ALL DONE
003652  001401                          BEQ     TS35
003654                          REG4E:
003654  104000                          EMT                     ;FAILURE WITH R4

                                ;*****************************************************************************
                                ;TEST 35        TEST IF R4 CAN HOLD A ZERO IN ALL BITS
                                ;*****************************************************************************
003656                          TS35:
003656  012704  177776                  MOV     #-2,R4          ;SET ALL ONES IN R4 EXCEPT FOR BIT 0
003662  012700  177757                  MOV     #-21,R0         ;SET BIT COUNTER
003666  000261                          SEC                     ;SET C-BIT
003670  005200                  REG4A:  INC     R0              ;INCREMENT BIT COUNTER
003672  001405                          BEQ     R4ERR           ;BR TO ERROR HALT IF COUNTER=0
003674  006104                          ROL     R4              ;ROTATE 1 POSITION
003676  103774                          BCS     REG4A           ;CONTINUE UNTIL C-BIT IS CLEAR
003700  022704  177777                  CMP     #-1,R4          ;CHECK DATA
003704  001401                          BEQ     TS36
003706                          R4ERR:
003706  104000                          EMT                     ;FAILURE WITH R4


                                ;*****************************************************************************
                                ;TEST 36        TEST IF R5 CAN HOLD A ONE IN ALL BITS
                                ;*****************************************************************************
003710                          TS36:
003710  012705  000001                  MOV     #1,R5           ;SET BIT 0
003714  012700  177757                  MOV     #-21,R0         ;SET BIT COUNTER
003720  000241                          CLC                     ;CLEAR C-BIT
003722  005200                  REG5:   INC     R0              ;INCREMENT BIT COUNTER
003724  001403                          BEQ     REG5E           ;BR TO ERROR HALT IF BIT IS LOST
003726  006105                          ROL     R5              ;ROTATE 1 POSITION
003730  103374                          BCC     REG5            ;ALL DONE
003732  001401                          BEQ     TS37
003734                          REG5E:
003734  104000                          EMT                     ;FAILURE WITH R5

                                ;*****************************************************************************
                                ;TEST 37        TEST IF R5 CAN HOLD A ZERO IN ALL BITS
                                ;*****************************************************************************
003736                          TS37:
003736  012705  177776                  MOV     #-2,R5          ;SET ALL ONES IN R5 EXCEPT FOR BIT 0
003742  012700  177757                  MOV     #-21,R0         ;SET BIT COUNTER
003746  000261                          SEC                     ;SET C-BIT
003750  005200                  REG5A:  INC     R0              ;INCREMENT BIT COUNTER
003752  001405                          BEQ     R5ERR           ;BR TO ERROR HALT IF COUNTER=0
003754  006105                          ROL     R5              ;ROTATE 1 POSITION
003756  103774                          BCS     REG5A           ;CONTINUE UNTIL C-BIT IS C;EAR
003760  022705  177777                  CMP     #-1,R5          ;CHECK DATA
003764  001401                          BEQ     TS40
003766                          R5ERR:
003766  104000                          EMT                     ;FAILURE WITH R5

                                ;*****************************************************************************
                                ;TEST 40        TEST IF R6 CAN HOLD A ONE IN ALL BITS
```

```
                                        ;*************************************************************************
003770                                  TS40:
003770  012706  000001                          MOV     #1,R6                   ;SET BIT 0
003774  012700  177757                          MOV     #-21,R0                 ;SET BIT COUNTER
004000  000241                                  CLC                             ;CLEAR C-BIT
004002  005200                          REG6:   INC     R0                      ;INCREMENT BIT COUNTER
004004  001403                                  BEQ     REG6E                   ;BR TO ERROR HALT IF BIT IS LOST
004006  006106                                  ROL     R6                      ;ROTATE 1 POSITION
004010  103374                                  BCC     REG6                    ;ALL DONE
004012  001401                                  BEQ     TS41
004014                                  REG6E:
004014  104000                                  EMT                             ;FAILURE WITH R6

                                        ;*************************************************************************
                                        ;TEST 41        TEST IF R6 CAN HOLD A ZERO IN ALL BITS

                                        ;*************************************************************************
004016                                  TS41:
004016  012706  177776                          MOV     #-2,R6                  ;SET ALL ONES IN R6 EXCEPT FOR BIT 0
004022  012700  177757                          MOV     #-21,R0                 ;SET BIT COUNTER
004026  000261                                  SEC                             ;SET C-BIT
004030  005200                          REG6A:  INC     R0                      ;INCREMENT BIT COUNT
004032  001405                                  BEQ     R6ERR                   ;BR TO ERROR HALT IF COUNTER=0
004034  006106                                  ROL     R6                      ;ROTATE 1 POSITION
004036  103774                                  BCS     REG6A                   ;CONTINUE UNTIL C-BIT IS CLEAR
004040  022706  177777                          CMP     #-1,R6                  ;CHECK DATA
004044  001401                                  BEQ     TS42
004046                                  R6ERR:
004046  104000                                  EMT                             ;FAILURE WITH R6

                                        ;*************************************************************************
                                        .SBTTL  PSW TESTS
                                        ;
                                        ;       THE PSW TESTS ARE USED TO VERIFY THAT VARIOUS DATA
                                        ;PATTERNS CAN BE SUCCESSFULLY HELD IN THE PSW AND THAT THE
                                        ;PSW ADDRESSING LOGIC IS FUNCTIONING.  MOVE AND COMPARE INSTRUCTIONS
                                        ;ARE USED TO TEST THAT THE PSW CAN HOLD VARIOUS DATA PATTERNS.
                                        ;EACH DATA PATTERN IS MOVED AND TESTED IN A SMALL LOOP CONVENIENT FOR
                                        ;SCOPING.
                                        ;       THE PSW REGISTER IS TESTED, THE CC INPUTS ARE TESTED
                                        ;LATER IN THE MICROCODE TESTS.  SETTING OF THE T-BIT BY THE
                                        ;TEST PATTERNS IS PURPOSELY AVOIDED. TESTING OF THE
                                        ;T-BIT TRAP CIRCUITRY IS LEFT FOR THE TRAP TEST.
                                        ;
                                        ;*************************************************************************
                                        ;TEST 42        TEST IF PSW WILL HOLD ZEROES
                                        ;*************************************************************************
004050                                  TS42:
004050  012706  001000                          MOV     #STBOT,R6
004054  012737  000000  177776                  MOV     #0,@#PS                 ;SET PSW TO ZERO
004062  005737  177776                          TST     @#PS                    ;SUCCESSFUL
004066  001401                                  BEQ     TS43
004070  104000                                  EMT                             ;PSW NOT 0

                                        ;*************************************************************************
                                        ;TEST 43        TEST IF PSW WILL HOLD ONES AND ZEROES
                                        ;*************************************************************************
```

```
004072                           TS43:
004072  012737  000252  177776          MOV     #252,@#PS       ;MOVE ALT. ONES AND ZEROES TO PSW
004100  023727  177776  000252          CMP     @#PS,#252       ;SUCCESSFUL?
004106  001401                          BEQ     TS44
004110  104000                          EMT                     ;PSW NOT 252

                                 ;*************************************************************************
                                 ;TEST 44        TEST IF PSW (EXCEPT T-BIT) WILL HOLD ZEROES AND ONES
                                 ;*************************************************************************
004112                           TS44:
004112  012737  000105  177776          MOV     #105,@#PS       ;MOVE ALT. ONES AND ZEROES TO PSW
004120  023727  177776  000105          CMP     @#PS,#105       ;SUCCESSFUL?
004126  001401                          BEQ     TS45
004130  104000                          EMT                     ;PSW NOT 105

                                 ;*************************************************************************
                                 ;TEST 45        TEST IF PSW (EXCEPT T-BIT) WILL HOLD ALL ONES
                                 ;*************************************************************************
004132                           TS45:
004132  012737  000357  177776          MOV     #357,@#PS       ;MOVE ONES TO PSW
004140  023727  177776  000357          CMP     @#PS,#357       ;SUCCESSFUL
004146  001401                          BEQ     TS46
004150  104000                          EMT                     ;PSW NOT 357
                                 ;*************************************************************************
                                 .SBTTL  MICROCODE TESTS
                                 ;
                                 ;       THE TEST EXERCISES  BRANCHES IN THE MICROCODE BY
                                 ;TESTING AT LEAST ONE INSTRUCTION FROM EVERY CLASS OF INSTRUCTION IN
                                 ;ALL POSSIBLE MODES.  FOR EXAMPLE, TO TEST THE SINGLE OPERAND INSTRUCTIONS,
                                 ;AT LEAST ONE SINGLE OPERAND INSTRUCTION IS VERIFIED IN ALL UNIQUE
                                 ;ADDRESSING MODES.  BYTE MODES ARE ALSO TESTED.  AS EACH NEW
                                 ;MODE IS INTRODUCED THE SAME INSTRUCTION IS TRIED AND TESTED IN
                                 ;A SMALL LOOP CONVENIENT FOR SCOPING.  THE TEST IS SET UP USING
                                 ;ONLY INSTRUCTIONS AND ADDRESSING MODES WHICH HAVE BEEN PREVIOUSLY
                                 ;VERIFIED.
                                 ;       IF THESE TESTS FAIL, CHECK THE RESULTS FOR A CLUE TO THE
                                 ;FAULT.
                                 ;
                                 ;*************************************************************************



                                 ;*************************************************************************
                                 ;
                                 ;       THE CLR INSTRUCTION IS USED TO INTRODUCE EACH ADDRESSING
                                 ;MODE WITH THE SINGLE OPERAND INSTRUCTION.  FOLLOWING THE SEQUENCE CHECK,
                                 ;THE CLR INSTRUCTION IS EXECUTED AND A BRANCH TEST IS EXECUTED WHICH
                                 ;CHECKS THAT THE Z-BIT WAS PROPERLY SET.  THIS TEST CAN CHECK IR DECODE
                                 ;AND MICROCODE FOR SOP INSTRUCTIONS WITH MODE 0.  FOLLOWING THIS TEST
                                 ;SEVERAL OTHER SOP INSTRUCTIONS ARE INTRODUCED WITH MODE 0.  THESE
                                 ;INSTRUCTIONS MAINPULATE DATA AND SERVE TO CHECK THE DATA RESULTS
                                 ;OF THE SOP INSTRUCTIONS IN THIS TEST.  THE DATA IN THIS TEST IS
                                 ;OPERATED ON BY EACH INSTRUCTION WITHOUT REINITIALIZING.
                                 ;
                                 ;*************************************************************************
                                 ;TEST 46        TEST MODE 0 USING SOP INST.
```

```
                        ;*****************************************************************************
004152                  TS46:
004152  005000                  CLR     R0              ;TRY THE CLEAR INST.
004154  001401                  BEQ     SOP0A
004156  104000                  EMT                     ;CLR DID NOT SET Z-BIT
004160  005200          SOP0A:  INC     R0              ;TRY THE INCREMENT INST.
004162  005100                  COM     R0              ;TRY COMPLEMENT
004164  005200                  INC     R0
004166  100401                  BMI     SOP0B
004170  104000                  EMT                     ;NEGATE DID NOT SET N-BIT
004172  005100          SOP0B:  COM     R0              ;TRY COMPLEMENT INST.
004174  001401                  BEQ     TS47
004176  104000                  EMT                     ;CUMMULATIVE RESULT OF CLR,INC,NEG AND COM INSTS. FAILED


                        ;*****************************************************************************
                        ;
                        ;       THIS TEST INTRODUCES THE REMAINING SOP INSTRUCIONS AND TESTS
                        ;THEM IN MODE 0.  THE PURPOSE IS TO PROVIDE A BASELINE OF
                        ;INSTRUCTIONS FOR USE IN THE SUBSEQUENT TESTS.  SINCE THE MICROCODE FOR
                        ;THESE INSTRUCTIONS IS IDENTICAL TO THAT ALREADY TESTED, ANY TROUBLE
                        ;SHOOTING EFFORTS SHOULD BE AIMED AT THE ACTUAL IR DECODE AND ALU
                        ;FUNCTIONING.
                        ;
                        ;*****************************************************************************
                        ;TEST 47        TEST REMAINDER OF SOP INSTS IN MODE 0
                        ;*****************************************************************************
004200                  TS47:
004200  005000                  CLR     R0              ;INITIALIZE
004202  005300                  DEC     R0              ;TRY DECREMENT INST.
004204  100401                  BMI     SOP0C
004206  104000                  EMT                     ;N-BIT NOT SET ON DEC
004210  000261          SOP0C:  SEC                     ;INITIALIZE CARRY
004212  005500                  ADC     R0              ;TRY ADD CARRY INST
004214  001007                  BNE     SOP0D
004216  000261                  SEC                     ;INITIALIZE CARRY
004220  005600                  SBC     R0              ;TRY SUBTRACT-CARRY INST
004222  100004                  BPL     SOP0D
004224  005100                  COM     R0
004226  005200                  INC     R0
004230  005300                  DEC     R0
004232  001401                  BEQ     TS50
004234                  SOP0D:
004234  104000                  EMT                     ; CUMMULATIVE RESULT OF ADC,SBC,COM,INC AND DEC INSTS. F

                        ;*****************************************************************************
                        ;
                        ;       THIS TEST INTRODUCES THE BYTE CONTROL LOGIC OF THE PROCESSOR.
                        ;THE MODE 0 BYTE MICROCODE IS TESTED.  THE METHOD AND SEQUENCE
                        ;OF TESTING IS THE SAME AS THAT USED IN THE SOP MODE 0 TESTS.
                        ;
                        ;*****************************************************************************
                        ;TEST 50        TEST MODE 0 EVEN BYTE USING SOP INST
                        ;*****************************************************************************
004236                  TS50:
004236  105000                  CLRB    R0              ;TRY CLEARING EVEN BYTE OF REGISTER
```

```
004240  001401            BEQ     SOPB0A
004242  104000            EMT                     ;CLRB DID NOT SET Z-BIT
004244  105100   SOPB0A:  COMB    R0              ;TRY SETTING EVEN BYTE OF REGISTER
004246  100002            BPL     SOPB0B
004250  105200            INCB    R0              ;TRY INCREMENTING EVEN BYTE OF REGISTER>>
004252  001401            BEQ     TS51
004254           SOPB0B:
004254  104000            EMT                     ;TEST CUMMULATIVE RESULT OF ABOVE BYTE INST.

                 ;*************************************************************************
                 ;
                 ;        THIS TEST USES THE CLR INSTRUCTION TO INTRODUCE AND TEST
                 ;SINGLE OPERAND MODE 1 INSTRUCTIONS.  AGAIN, THE CLR INSTRUCTION
                 ;IS USED TO INTRODUCE THE MICROCODE AND TO TEST THAT THE PROPER
                 ;CONDITION CODES ARE SET.  OTHER SOP INSTRUCTIONS ARE USED TO MANIPULATE
                 ;COMMON DATA TO VERIFY THAT THE CORRECT DATA IS PRODUCED.
                 ;
                 ;*************************************************************************
                 ;TEST 51         TEST MODE 1 USING SOP INST.
                 ;*************************************************************************
004256           TS51:
004256  005000            CLR     R0              ;INITIALIZE R0
004260  005010            CLR     (R0)            ;TRY CLEAR INST W/MODE 1
004262  001401            BEQ     SOP1A
004264  104000            EMT                     ;CLR DID NOT SET Z-BIT
004266  005310   SOP1A:   DEC     (R0)            ;TRY DECREMENT INST W/MODE 1
004270  100003            BPL     SOP1B
004272  000261            SEC                     ;INITIALIZE CARRY
004274  005510            ADC     (R0)            ;TRY ADD-CARRY W/MODE 1
004276  001401            BEQ     TS52
004300           SOP1B:
004300  104000            EMT                     ;TEST CUMMULATIVE RESULT OF ABOVE INST

                 ;*************************************************************************
                 ;
                 ;        THIS TEST VERIFIES THE BYTE INSTRUCTION MICROCODE FOR MODE 1
                 ;SINGLE OPERAND INSTRUCTIONS.
                 ;        THIS IS THE FIRST PLACE THE SIGN EXTEND LOGIC IS EXERCISED
                 ;AND VERIFIED.
                 ;
                 ;*************************************************************************
                 ;TEST 52         TEST MODE 1 EVEN BYTE USING SOP INST
                 ;*************************************************************************
004302           TS52:
004302  005000            CLR     R0              ;INITIALIZE R0
004304  005010            CLR     (R0)            ;INITIALIZE LOC. 0
004306  005110            COM     (R0)
004310  105010            CLRB    (R0)            ;TRY TO CLEAR BYTE 0
004312  001401            BEQ     SOPB1A
004314  104000            EMT                     ;CLRB DID NOT SET Z-BIT
004316  005210   SOPB1A:  INC     (R0)            ;INCREMENT TO TEST WORD
004320  100005            BPL     SOPB1B
004322  105110            COMB    (R0)            ;COMPLEMENT:  ODD BYTE = 376
004324  105210            INCB    (R0)            ;INC:  ODD BYTE = 377
004326  100002            BPL     SOPB1B
004330  105210            INCB    (R0)            ;INCREMENT ODD BYTE=0
```

```
004332  001401                  BEQ     TS53
004334                  SOPB1B:
004334  104000                  EMT                     ;CHECK CUMMULATIVE RESULT OF ABOVE INST


                        ;*******************************************************************************
                        ;
                        ;       THIS TEST VERIFIES THAT SINGLE OPERAND BYTE INSTRUCTIONS WILL
                        ;FUNCTION CORRECTLY FOR ODD BYTES.
                        ;       THIS IS THE FIRST TIME THAT ADDRESS LINE 0 HAS BEEN
                        ;EXERCISED.  CHECKS ARE MADE THAT THE PROPER BYTE IS MODIFIED AND
                        ;THE CONDITION CODES ARE CHECKED.  IT IS ALSO VERIFIED THAT THE UNADDRESSED
                        ;BYTE IS NOT ALTERED BY THE INSTRUCTION.
                        ;
                        ;*******************************************************************************
                        ;TEST 53        TEST MODE 1 ODD BYTE USING SOP INST
                        ;*******************************************************************************
004336                  TS53:
004336  005000                  CLR     R0              ;INITIALIZE R0
004340  005010                  CLR     (R0)            ;INITIALIZE LOC. 0
004342  005110                  COM     (R0)
004344  005200                  INC     R0              ;R0=ODD BYTE
004346  105010                  CLRB    (R0)            ;TRY TO CLEAR BYTE 1
004350  001401                  BEQ     SOPB1C
004352  104000                  EMT                     ;CLRB DID NOT SET Z-BIT
004354  005300          SOPB1C: DEC     R0              ;R0=WORD ADDR.
004356  005210                  INC     (R0)            ;INCREMENT TO TEST WORD
004360  005200                  INC     R0              ;R0=ODD BYTE
004362  105110                  COMB    (R0)            ;TRY TO COMPLEMENT BYTE 1
004364  105210                  INCB    (R0)
004366  100002                  BPL     SOPB1D
004370  105210                  INCB    (R0)            ;TRY TO INCREMENT BYTE 1
004372  001401                  BEQ     TS54
004374                  SOPB1D:
004374  104000                  EMT                     ;TEST CUMMULATIVE RESULT OF ABOVE INST.

                        ;*******************************************************************************
                        ;
                        ;       THIS TEST VERIFIES MODE 2 SINGLE-OPERAND INSTRUCTIONS.  PREVIOUSLY
                        ;TESTED INSTRUCTIONS ARE USED TO SET A POINTER IN R0 TO LOC. 400.
                        ;LOC. 400 IS INITIALIZED TO -1 BEFORE A CLR MODE 2 IS EXECUTED.
                        ;       THEN R0 IS DECREMENTED BY TWO TO AGAIN POINT TO 400 BEFORE EACH
                        ;OF SEVERAL MODE 2 INSTRUCTIONS ARE USED TO VERIFY THE DATA RESULTS OF
                        ;THE TEST.  THIS PROCEDURE ALSO VERIFIES THE PROPER INCREMENTING OF THE
                        ;REGISTER.
                        ;
                        ;*******************************************************************************
                        ;TEST 54        TEST MODE 2 USING SOP INST.
                        ;*******************************************************************************
004376                  TS54:
004376  005000                  CLR     R0              ;SET R0=400
004400  105100                  COMB    R0
004402  005200                  INC     R0
004404  005010                  CLR     (R0)            ;CLEAR 400
004406  005110                  COM     (R0)            ;INITIALIZE: 400=-1
004410  005020                  CLR     (R0)+           ;TRY CLEARING WITH MODE 2
```

```
004412  001401                  BEQ     SOPZA
004414  104000                  EMT                     ;CLR INST DID NOT SET Z-BIT
004416  005300          SOPZA:  DEC     R0              ;RESET R0
004420  005300                  DEC     R0
004422  005120                  COM     (R0)+           ;TRY COMPLEMENTING WITH MODE 2
004424  100004                  BPL     SOP2B
004426  005300                  DEC     R0              ;RESET R0
004430  005300                  DEC     R0
004432  005220                  INC     (R0)+           ;TRY INCREMENTING WITH MODE 2
004434  001401                  BEQ     TS55
004436                  SOP2B:
004436  104000                  EMT                     ;CHECK CUMMULATIVE RESULT OF ABOVE INST

                        ;**************************************************************************
                        ;
                        ;       THIS TEST VERIFIES MODE 2 SINGLE OPERAND INSTRUCTIONS WHICH
                        ;ADDRESS EVEN BYTES.  R0 IS SET TO 400 AND USED TO INITIALIZE LOCATION
                        ;400 TO -1.  CLRB INSTRUCTION IS THEN EXECUTED ON BYTE 400 WITH
                        ;MODE 2.
                        ;       R0 IS THEN DECREMENTED BEFORE EACH OF SEVERAL MODE 2 INSTRUCTIONS
                        ;WHICH ARE USED TO VERIFY THE DATA RESULTS OF THE TEST.  THIS PROCEDURE ALSO
                        ;VERIFIES THE PROPER INCREMENTING OF THE REGISTER.
                        ;
                        ;**************************************************************************
                        ;TEST 55        TEST MODE 2 EVEN BYTE USING SOP INST.
                        ;**************************************************************************
004440                  TS55:
004440  005000                  CLR     R0              ;SET R0=400
004442  105100                  COMB    R0
004444  005200                  INC     R0
004446  005010                  CLR     (R0)            ;CLEAR 400
004450  005110                  COM     (R0)            ;INITIALIZE: 400=-1
004452  105020                  CLRB    (R0)+           ;TRY TO CLEAT 400 W/MODE 2
004454  001401                  BEQ     SOPB2A
004456  104000                  EMT                     ;CLR DID NOT SET Z-BIT
004460  005300          SOPB2A: DEC     R0              ;RESULT R0=400
004462  005210                  INC     (R0)            ;INC 400 TO TEST WORD
004464  105110                  COMB    (R0)
004466  105220                  INCB    (R0)+           ;TRY TO INC EVEN BYTE
004470  100003                  BPL     SOPB2B
004472  005300                  DEC     R0              ;RESET R0=400
004474  105220                  INCB    (R0)+           ;TRY INCREMENT OF EVEN BYTE
004476  001401                  BEQ     TS56
004500                  SOPB2B:
004500  104000                  EMT                     ;TEST CUMMULATIVE RESULT OF ABOVE INST.

                        ;**************************************************************************
                        ;
                        ;       THIS TEST FOLLOWS THE SAME PROCEDURE DESCRIBED IN THE PREVIOUS
                        ;TEST.  HERE, THE BYTE INSTRUCTION IS USED TO ADDRESS AN ODD BYTE.
                        ;
                        ;**************************************************************************
                        ;TEST 56        TEST MODE 2 ODD BYTE USING SOP INST.
                        ;**************************************************************************
004502                  TS56:
004502  005000                  CLR     R0              ;SET R0=400
```

```
004504  105100                  COMB    R0
004506  005200                  INC     R0
004510  005010                  CLR     (R0)            ;CLEAR LOC 400
004512  005110                  COM     (R0)            ;INITIALIZE: 400=-1
004514  005200                  INC     R0              ;R0=ODD BYTE
004516  105020                  CLRB    (R0)+           ;TRY TO CLEAR ODD BYTE
004520  001401                  BEQ     SOPB2C
004522  104000                  EMT                     ;CLRB DID NOT SET Z-BIT
004524  005300          SOPB2C: DEC     R0              ;R0=WORD ADDR.
004526  005300                  DEC     R0
004530  005220                  INC     (R0)+           ;INCREMENT WORD
004532  005300                  DEC     R0              ;POINT TO ODD BYTE
004534  105110                  COMB    (R0)            ;COMPLEMENT ODD BYTE
004536  105220                  INCB    (R0)+           ;TRY TO INCREMENT ODD BYTE
004540  100003                  BPL     SOPB2D
004542  005300                  DEC     R0              ;RESET R0 TO ODD BYTE
004544  105220                  INCB    (R0)+           ;TRY TO INCREMENT ODD BYTE
004546  001401                  BEQ     TS57
004550                  SOPB2D:
004550  104000                  EMT                     ;TEST CUMMULATIVE RESULT OF ABOVE INST.

                        ;*************************************************************************
                        ;
                        ;       THESE TESTS CHECK THE NEGATE INSTRUCTION IN ALL MODES.  PREVIOUSLY
                        ;TESTED SINGLE-OPERAND INSTRUCTIONS ARE USED TO TEST THE NEGATE INSTRUCTION.
                        ;
                        ;*************************************************************************
                        ;TEST 57        TEST MODE 0 USING NEGATE INSTRUCTION
                        ;*************************************************************************
004552                  TS57:
004552  005000                  CLR     R0              ;SET R0=0
004554  005200                  INC     R0              ;    R0=1
004556  005400                  NEG     R0              ;TRY NEGATE MODE 0:  R0=-1
004560  100003                  BPL     NEG00           ;CC=1001?
004562  001402                  BEQ     NEG00
004564  102401                  BVS     NEG00
004566  103401                  BCS     NEG01
004570                  NEG00:
004570  104000                  EMT                     ;NEGATE DID NOT SET CC'S CORRECTLY

004572  005200          NEG01:  INC     R0              ;TEST DATA RESULT
004574  001401                  BEQ     NEG02
004576  104000                  EMT                     ;DATA RESULT OF NEGATE INCORRECT

004600  105100          NEG02:  COMB    R0              ;R0=377
004602  105400                  NEGB    R0              ;R0=1
004604  100403                  BMI     NEG03           ;CC=0001?
004606  001402                  BEQ     NEG03
004610  102401                  BVS     NEG03
004612  103401                  BCS     NEG04
004614                  NEG03:
004614  104000                  EMT                     ;NEGB DID NOT SET CC'S CORRECTLY
004616  005300          NEG04:  DEC     R0              ;TEST DATA RESULT
004620  001401                  BEQ     TS60
004622  104000                  EMT                     ;DATA RESULT OF NEGB INCORRECT
                        ;*************************************************************************
```

```
                                    ;TEST 60        TEST MODE 1 USING NEGATE INST.
                                    ;*****************************************************************************
        004624                      TS60:
        004624  005000                      CLR     R0                      ;POINT TO LOC. 0
        004626  005010                      CLR     (R0)                    ;CLEAR LOC. 0
        004630  005210                      INC     (R0)                    ;LOC. 0=1
        004632  005410                      NEG     (R0)                    ;TRY NEG. LOC. 0=-1
        004634  100003                      BPL     NEG10                   ;CC=1001
        004636  001402                      BEQ     NEG10
        004640  102401                      BVS     NEG10
        004642  103401                      BCS     NEG11
        004644                      NEG10:
        004644  104000                      EMT                             ;NEGATE DID NOT SET CC'S CORRECTLY

        004646  005237  000000      NEG11:  INC     @#0                     ;TEST DATA RESULT
        004652  001401                      BEQ     NEG12
        004654  104000                      EMT                             ;DATA RESULT OF NEGATE INCORRECT
        004656  105110              NEG12:  COMB    (R0)                    ;LOC. 0=377
        004660  105410                      NEGB    (R0)                    ;TRY NEGB LOC. 0=1
        004662  100403                      BMI     NEG13                   ;CC=0001?
        004664  001402                      BEQ     NEG13
        004666  102401                      BVS     NEG13
        004670  103401                      BCS     NEG14
        004672                      NEG13:
        004672  104000                      EMT                             ;NEGB DID NOT SET CC'S CORRECTLY
        004674  005337  000000      NEG14:  DEC     @#0                     ;TEST DATA RESULT
        004700  001401                      BEQ     TS61
        004702  104000                      EMT                             ;DATA RESULT OF NEGB INCORRECT
                                    ;*****************************************************************************
                                    ;TEST 61        TEST MODE 2 USING NEGATE INSTRUCTION
                                    ;*****************************************************************************
        004704                      TS61:
        004704  005000                      CLR     R0                      ;POINT TO LOC. 0
        004706  005010                      CLR     (R0)                    ;CLEAR LOC. 0
        004710  005210                      INC     (R0)                    ;LOC. 0=1
        004712  005420                      NEG     (R0)+                   ;TRY NEG.: LOC. 0=-1
        004714  100003                      BPL     NEG20                   ;CC=1001?
        004716  001402                      BEQ     NEG20
        004720  102401                      BVS     NEG20
        004722  103401                      BCS     NEG21
        004724                      NEG20:
        004724  104000                      EMT                             ;NEGATE DID NOT SET CC'S CORRECTLY
        004726  105300              NEG21:  DECB    R0                      ;R0=LOC. 0
        004730  105300                      DECB    R0
        004732  105420                      NEGB    (R0)+                   ;BYTE 0=1   R0=1
        004734  105420                      NEGB    (R0)+                   ;BYTE 1=1   R0=2
        004736  105340                      DECB    -(R0)                   ;R0=1  LOC. 0=01
        004740  005300                      DEC     R0                      ;R0=0
        004742  001401                      BEQ     NEG22
        004744  104000                      EMT                             ;REGISTER NOT INCREMENTED CORRECTLY
        004746  005337  000000      NEG22:  DEC     @#0                     ;LOC. 0=0
        004752  001401                      BEQ     TS62
        004754  104000                      EMT                             ;NEG BYTE INSTRUCTIONS FAILED

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

```
                                ;       THIS TEST VERIFIES MODE 3 SINGLE OPERAND INSTRUCTIONS.  IT
                                ;USES LOCATION 0 AS ITS TARGET DATA.  A TABLE LOCATED AT LOC. 400
                                ;THRU 402 IS USED TO SUPPLY THE ADDRESS OF LOCATION 0 TO THE
                                ;INSTRUCTIONS UNDER TEST.
                                ;       R0 IS SET TO 400, THE START OF THE ADDRESS TABLE, AND A CLR
                                ;INSTRUCTION IS EXECUTED WITH MODE 3 TO CLEAR LOC. 0.  THEN R0
                                ;IS DECREMENTED BY TWO AND TWO OTHER MODE 3 INSTRUCTIONS OPERATE ON
                                ;LOC. 0 TO VERIFY THE DATA RESULTS OF THE TEST.  THE PROPER INCREMENTING
                                ;OF THE REGISTER IS ALSO VERIFIED IN THIS MANNER.
                                ;       IF A FAILURE IS DETECTED BE SURE TO VERIFY THAT THE TABLE
                                ;(LOC. 400-402) HAS THE PROPER VALUES (0).
                                ;
                                ;***************************************************************************
                                ;TEST 62        TEST MODE 3 USING SOP INST.
                                ;***************************************************************************
004756                          TS62:
004756  005000                          CLR     R0              ;SET R0=400
004760  105100                          COMB    R0
004762  005200                          INC     R0
004764  005010                          CLR     (R0)            ;CLEAR LOC 400
004766  005030                          CLR     @(R0)+          ;TRY TO CLEAR LOC 0 USING MODE 3 ;R0=402
004770  001401                          BEQ     SOP3A
004772  104000                          EMT                     ;CLR DID NOT SET Z-BIT
004774  005300                  SOP3A:  DEC     R0              ;RESET R0=400
004776  005300                          DEC     R0
005000  005130                          COM     @(R0)+          ;TRY TO COMPLEMENT LOC 0 OF MODE 3  ;R0=402
005002  100002                          BPL     SOP3B
005004  005230                          INC     @(R0)+          ;TRY TO INCREMENT LOC 0 W/MODE 3  ;R0=404
005006  001401                          BEQ     TS63
005010                          SOP3B:
005010  104000                          EMT                     ;CUMMULATIVE RESULT OF ABOVE INST FAILED

                                ;***************************************************************************
                                ;
                                ;       THIS TEST VERIFIES MODE 3 SINGLE OPERAND BYTE INSTRUCTIONS
                                ;WHICH ADDRESS EVEN BYTES.  AGAIN, THE TARGET LOCATION 0 IS USED
                                ;AND THE SAME TABLE AT 400 IS EMPLOYED.
                                ;       AFTER POINTING R4 TO THE TABLE (400) AND SETTING LOCATION
                                ;0 TO -1, A CLRB INSTRUCTION IS USED TO CLEAR BYTE 0.
                                ;       SEVERAL OTHER MODE 3 INSTRUCTIONS ARE THEN USED WITH THE TABLE
                                ;TO VERIFY THE DATA RESULTS AND THE PROPER INCREMENTING OF THE REGISTER.
                                ;IF A FAILURE IS DETECTED, BE SURE THAT THE TABLE (LOCATION 400-402) HAS
                                ;THE PROPER VALUES (0).
                                ;
                                ;***************************************************************************
                                ;TEST 63        TEST MODE 3 EVEN BYTE USING SOP INST.
                                ;***************************************************************************
005012                          TS63:
005012  005004                          CLR     R4              ;SET R4=400
005014  105104                          COMB    R4
005016  005204                          INC     R4
005020  005000                          CLR     R0              ;INITIALIZE LOC. 0=-1
005022  005010                          CLR     (R0)
005024  005110                          COM     (R0)            ;LOC. 0=-1
005026  105034                          CLRB    @(R4)+          ;TRY TO CLEAR EVEN BYTE ;LOC. 0=177400  R4=402
005030  001401                          BEQ     SOPB3A
```

```
005032  104000                  EMT                     ;CLRB DID NOT SET Z-BIT
005034  005304          SOPB3A: DEC     R4              ;RESET POINTER  R4=400
005036  005304                  DEC     R4
005040  005234                  INC     @(R4)+          ;TRY INCREMENTING WORD  LOC.0=177401  R4=402
005042  100006                  BPL     SOPB3B
005044  105434                  NEGB    @(R4)+          ;TRY TO NEGATE EVEN BYTE ;LOC.0=-1 R4=404
005046  100004                  BPL     SOPB3B
005050  005304                  DEC     R4              ;R4=402
005052  005304                  DEC     R4
005054  105234                  INCB    @(R4)+          ;TRY TO INCREMENT EVEN BYTE ;LOC. 0=17400
005056  001401                  BEQ     TS64
005060                  SOPB3B:
005060  104000                  EMT                     ;CUMMULATIVE RESULT OF ABOVE INST FAILED

                        ;******************************************************************************* **
                        ;
                        ;       THIS TEST VERIFIES MODE 3 SINGLE OPERAND BYTE INSTRUCTIONS
                        ;WHICH ADDRESS ODD BYTES.  THE TARGET IS BYTE 1.  A TABLE AT
                        ;LOC. 400-406 IS USED.  R0 SERVES AS THE TABLE POINTER.
                        ;       R0 IS INITIALIZED TO 400.  LOC. 0 IS SET TO -1 USING THE
                        ;FIRST TWO TABLE ENTRIES.  A CLRB MODE 3 IS EXECUTED ON BYTE 1 USING
                        ;TABLE ADDRESS AT 404.  R0 IS DECREMENTED TO 402 AND SEVERAL SOP
                        ;MODE 3 INSTRUCTIONS ARE USED TO VERIFY DATA RESULTS AND PROPER
                        ;REGISTER INCREMENTING.
                        ;       THE TABLE (400-406) SHOULD CONTAIN 0,0,1,1 BEFORE AND
                        ;AFTER THE TEST IS RUN.

                        ;**********************************************************************************
                        ;TEST 64        TEST MODE 3 ODD BYTE USING SOP INST.
                        ;**********************************************************************************
005062                  TS64:
005062  005000                  CLR     R0              ;SET R0=400
005064  105100                  COMB    R0
005066  005200                  INC     R0
005070  005030                  CLR     @(R0)+          ;INITIALIZE
005072  005130                  COM     @(R0)+          ;LOC 0=-1 R0=404
005074  105030                  CLRB    @(R0)+          ;TRY TO CLEAR ODD BYTE LOC. 0=377 R0=406
005076  001401                  BEQ     SOPB3C
005100  104000                  EMT                     ;CLRB DID NOT SET Z-BIT
005102  005300          SOPB3C: DEC     R0              ;RESET R0=402
005104  005300                  DEC     R0
005106  005300                  DEC     R0              ;POINT TO EVEN BYTE ADDR.
005110  005300                  DEC     R0
005112  005230                  INC     @(R0)+          ;INCREMENT WORD LOC. 0=400 R0=404
005114  105430                  NEGB    @(R0)+          ;TRY TO NEGATE ODD BYTE LOC. 0=177400 R0=406
005116  100002                  BPL     SOPB3D
005120  105230                  INCB    @(R0)+          ;TRY TO INCREMENT ODD BYTE LOC.0=0 R0=410
005122  001401                  BEQ     TS65
005124                  SOPB3D:
005124  104000                  EMT                     ;CUMMULATIVE RESULT OF ABOVE INSTS FAILED
                        ;**********************************************************************************
                        ;TEST 65        TEST MODE 3 USING NEGATE INSTRUCTION
                        ;**********************************************************************************
005126                  TS65:
005126  005000                  CLR     R0              ;R0=400
005130  105100                  COMB    R0
```

```
005132  005200                          INC     R0
005134  005010                          CLR     (R0)            ;LOC. 400=0
005136  005004                          CLR     R4              ;R4=0
005140  005014                          CLR     (R4)            ;LOC. 0=0
005142  005214                          INC     (R4)            ;LOC. 0=1
005144  005430                          NEG     @(R0)+          ;TRY NEGATE   LOC. 0=-1  R0=402
005146  100003                          BPL     NEG30           ;CC=1001?
005150  001402                          BEQ     NEG30
005152  102401                          BVS     NEG30
005154  103401                          BCS     NEG31
005156                          NEG30:
005156  104000                          EMT                     ;NEG DID NOT SET CC'S CORRECTLY
005160  005214                  NEG31:  INC     (R4)            ;LOC. 0=0
005162  001401                          BEQ     NEG32
005164  104000                          EMT                     ;DATA RESULT OF NEG INCORRECT
005166  105137  000001          NEG32:  COMB    @#1             ;LOC 0=177400
005172  005237  000000                  INC     @#0             ;LOC. 0=177401
005176  105430                          NEGB    @(R0)+          ;TRY NEGB LOC. 0=177777  R0=404
005200  100401                          BMI     NEG33
005202  104000                          EMT                     ;NEGB FAILED WITH EVEN BYTE
005204  105430                  NEG33:  NEGB    @(R0)+          ;TRY NEGB  LOC.0=777  R0=406
005206  100001                          BPL     NEG34
005210  104000                          EMT                     ;NEGB FAILED WITH ODD BYTE
005212  105137  000001          NEG34:  COMB    @#1             ;LOC. 0=177377
005216  105237  000001                  INCB    @#1             ;LOC. 0=177777
005222  005214                          INC     (R4)            ;LOC. 0=0
005224  001401                          BEQ     TS66
005226  104000                          EMT                     ;DATA RESULT OF NEGB'S INCORRECT

                                ;****************************************************************************
                                ;
                                ;       THIS TEST VERIFIES MODE 4 SINGLE OPERAND INSTRUCTIONS.
                                ;R0 IS SET TO 400. A CLR INSTRUCTION IS EXECUTED IN MODE 4 TO CLEAR
                                ;LOC. 376. R0 IS RESET TO 400 AND A COM INSTRUCTION USING MODE 4
                                ;COMPLEMENTS LOC.376.
                                ;       TWO INC INSTRUCTIONS AND A MODE 4 INSTRUCTION ARE EXECUTED
                                ;TO COMPLETE THE TEST.
                                ;****************************************************************************
                                ;TEST 66        TEST MODE 4 USING SOP INSTS
                                ;****************************************************************************
005230                          TS66:
005230  005000                          CLR     R0              ;SET R0=400
005232  105100                          COMB    R0
005234  005200                          INC     R0
005236  005040                          CLR     -(R0)           ;TRY TO CLEAR USING MODE 4
005240  001401                          BEQ     SOP4A
005242  104000                          EMT                     ;CLR DID NOT SET Z-BIT
005244  005200                  SOP4A:  INC     R0              ;RESET R0
005246  005200                          INC     R0
005250  005140                          COM     -(R0)           ;TRY TO COMPLEMENT USING MODE 4
005252  100004                          BPL     SOP4B
005254  005200                          INC     R0              ;MOVE POINTER
005256  005200                          INC     R0
005260  005240                          INC     -(R0)
005262  001401                          BEQ     TS67
005264                          SOP4B:
```

```
005264  104000                          EMT                     ;CHECK CUMMULATIVE RESULT OF ABOVE INST.

                                ;*******************************************************************************
                                ;
                                ;       THIS TEST VERIFIES MODE 5 SINGLE OPERAND INSTRUCTIONS.  IT
                                ;USES LOCATION 0 AS ITS TARGET DATA.  A TABLE LOCATED AT LOC. 372
                                ;THRU 374 IS USED TO SUPPLY THE ADDRESS OF LOCATION 0 TO THE
                                ;INSTRUCTIONS UNDER TEST.
                                ;       R0 IS SET TO 376, (THE START OF THE ADDRESS TABLE) +2,
                                ;AND A CLR INSTRUCTION IS EXECUTED WITH MODE 3 TO CLEAR
                                ;LOC. 0.  THEN R0 IS INCREMENTED BY TWO AND TWO OTHER MODE 3
                                ;INSTRUCTIONS OPERATE ON LOC. 0 TO VERIFY THE DATA RESULTS OF
                                ;THE TEST.  THE PROPER DECREMENTING OF THE REGISTER IS ALSO
                                ;VERIFIED IN THIS MANNER.
                                ;       IF A FAILURE IS DETECTED BE SURE TO VERIFY THAT THE TABLE
                                ;(LOC. 372 THRU 374) HAS THE PROPER VALUES (0).
                                ;
                                ;*******************************************************************************
                                ;TEST 67        TEST MODE 5 USING SOP INSTS
                                ;*******************************************************************************
005266                          TS67:
005266  012700  000370                  MOV     #370,R0         ;CLEAR LOCATION 370-376
005272  005020                          CLR     (R0)+           ;370
005274  005020                          CLR     (R0)+           ;372
005276  005020                          CLR     (R0)+           ;374
005300  005010                          CLR     (R0)            ;376
005302  005000                          CLR     R0              ;SET R0=376 (LOW BYTE)
005304  005020                          CLR     (R0)+
005306  105400                          NEGB    R0
005310  005050                          CLR     @-(R0)          ;TRY TO CLEAR LOC 0 W/MODE 5
005312  001401                          BEQ     SOP5A
005314  104000                          EMT                     ;CLR DID NOT SET Z-BIT
005316  005200                  SOP5A:  INC     R0              ;RESET R0
005320  005200                          INC     R0
005322  005150                          COM     @-(R0)          ;TRY TO COMPLEMENT LOC. 0 W/MODE 5
005324  100002                          BPL     SOP5B
005326  005250                          INC     @-(R0)          ;TRY TO INCREMENT LOC. 0 W/MODE 5
005330  001401                          BEQ     TS70
005332                          SOP5B:
005332  104000                          EMT                     ;TEST CUMMULATIVE RESULT OF ABOVE INSTS

                                ;*******************************************************************************
                                ;
                                ;       THIS TEST VERIFIES MODE 6 SINGLE OPERAND INSTRUCTIONS.  IT
                                ;USES LOCATION 0 AS ITS TARGET DATA.  R0 IS SET TO 400 USING
                                ;PREVIOUSLY TESTED INSTRUCTIONS AND A MODE 6 CLR INSTRUCTION IS
                                ;EXECUTED ON LOC. 0 USING R0 AND A -400 OFFSET.  COM AND INC
                                ;INSTRUCTIONS ARE THEN USED TO VERIFY THE DATA.
                                ;
                                ;*******************************************************************************
                                ;TEST 70        TEST MODE 6 USING SOP INSTS
                                ;*******************************************************************************
005334                          TS70:
005334  005000                          CLR     R0              ;SET R0=400
005336  105100                          COMB    R0
005340  005200                          INC     R0
```

```
005342  005060  177400            CLR     -400(R0)          ;TRY TO CLEAR LOCATION 0 W/MODE 6
005346  001401                    BEQ     SOP6A
005350  104000                    EMT                       ;CLR DID NOT SET Z-BIT
005352  005160  177400    SOP6A:  COM     -400(R0)          ;TRY TO COMPLEMENT LOCATION 0 W/MODE 6
005356  100003                    BPL     SOP6B
005360  005260  177400            INC     -400(R0)          ;TRY TO INCREMENT LOCATION 0 W/MODE 6
005364  001401                    BEQ     TS71
005366                    SOP6B:
005366  104000                    EMT                       ;TEST CUMMULATIVE RESULT OF ABOVE INSTS

                          ;*******************************************************************************
                          ;
                          ;       THIS TEST VERIFIES MODE 7 SINGLE OPERAND INSTRUCTIONS.  IT USES
                          ;THE POINTER TO LOC. 0 WHICH IS STORED AT LOC. 402.
                          ;       R0 IS SET TO 400 AND A MODE 7 CLR INSTRUCTION IS
                          ;EXECUTED WITH A +2 OFFSET TO CLEAR LOC. 0.
                          ;       SEVERAL OTHER MODE 7 INSTRUCTIONS ARE THEN USED ON THE COMMON
                          ;LOCATION TO VERIFY THE DATA RESULTS.
                          ;
                          ;*******************************************************************************
                          ;TEST 71        TEST MODE 7 USING SOP INST.
                          ;*******************************************************************************
005370                    TS71:
005370  005000                    CLR     R0                ;SET R0=400
005372  105100                    COMB    R0
005374  005200                    INC     R0
005376  005210                    INC     (R0)              ;R0=1
005400  005070  000002            CLR     @2(R0)            ;TRY TO CLEAR LOC. 0 W/MODE 7
005404  001401                    BEQ     SOP7A
005406  104000                    EMT                       ;CLR DID NOT SET Z-BIT
005410  005170  000002    SOP7A:  COM     @2(R0)            ;TRY TO COMPLEMENT LOC. 0 W/MODE 7
005414  100003                    BPL     SOP7B
005416  005270  000002            INC     @2(R0)            ;TRY TO INCREMENT LOC. 0 W/MODE 7
005422  001401                    BEQ     TS72
005424                    SOP7B:
005424  104000                    EMT                       ;TEST CUMMULATIVE RESULT OF ABOVE INSTS.

                          ;*******************************************************************************
                          ;TEST 72        TEST MODE 4 WITH NEGATE INSTRUCTION
                          ;*******************************************************************************
005426                    TS72:
005426  005000                    CLR     R0
005430  005010                    CLR     (R0)
005432  005120                    COM     (R0)+             ;LOC. 0=177777,  R0=2
005434  005440                    NEG     -(R0)             ;TRY NEGATE, LOC. 0=1
005436  100403                    BMI     NEG40             ;CC=0001?
005440  001402                    BEQ     NEG40
005442  102401                    BVS     NEG40
005444  103401                    BCS     NEG41
005446                    NEG40:
005446  104000                    EMT                       ;NEG DID NOT SET CC'S CORRECTLY
005450  005400            NEG41:  NEG     R0                ;TST R0 WITH A NEG.
005452  001401                    BEQ     NEG42
005454  104000                    EMT                       ;R0 NOT DECREMENTED PROPERLY
005456  005310            NEG42:  DEC     (R0)              ;TEST DTA RESULT OF NEG
005460  001401                    BEQ     TS73
```

```
005462  104000                  EMT                     ;DATA RESULT OF NEG INCORRECT
                        ;*******************************************************************************
                        ;TEST 73        TEST MODE 5 WITH NEGATE INSTRUCTION
                        ;*******************************************************************************
005464                  TS73:
005464  005000                  CLR     R0              ;R0=0
005466  005010                  CLR     (R0)            ;LOC. 0=0
005470  105100                  COMB    R0              ;R0=377
005472  005200                  INC     R0              ;R0=400
005474  005010                  CLR     (R0)            ;SET 400 = 0
005476  005004                  CLR     R4              ;R4=0
005500  005314                  DEC     (R4)            ;LOC. 0=177777
005502  005450                  NEG     @-(R0)          ;TRY NEGATE:  LOC. 0=1
005504  100403                  BMI     NEG50           ;CC=0001?
005506  001402                  BEQ     NEG50
005510  102401                  BVS     NEG50
005512  103401                  BCS     NEG51
005514                  NEG50:
005514  104000                  EMT                     ;NEG DID NOT SET CC'S CORRECTLY
005516  005314          NEG51:  DEC     (R4)
005520  001401                  BEQ     NEG52
005522  104000                  EMT                     ;DATA RESULT OF NEG INCORRECT
005524  105100          NEG52:  COMB    R0
005526  005300                  DEC     R0
005530  001401                  BEQ     TS74
005532  104000                  EMT                     ;REGISTER NOT DECREMENTED PROPERLY
                        ;*******************************************************************************
                        ;TEST 74        TEST MODE 6 WITH NEGATE
                        ;*******************************************************************************
005534                  TS74:
005534  005000                  CLR     R0              ;R0=0
005536  005004                  CLR     R4              ;R4=0
005540  105100                  COMB    R0              ;R0=377
005542  005014                  CLR     (R4)            ;LOC. 0=0
005544  105024                  CLRB    (R4)+           ;LOC. 0=177777,  R4=1
005546  105114                  COMB    (R4)            ;LOC. 0=177400
005550  005460  177401          NEG     -377(R0)        ;LOC. 0=400
005554  100403                  BMI     NEG60           ;CC=0001
005556  001402                  BEQ     NEG60
005560  102401                  BVS     NEG60
005562  103401                  BCS     NEG61
005564                  NEG60:
005564  104000                  EMT                     ;NEG DID NOT SET CC'S CORRECTLY
005566  105314          NEG61:  DECB    (R4)
005570  001401                  BEQ     TS75
005572  104000                  EMT                     ;DATA RESULT OF NEG INCORRECT
                        ;*******************************************************************************
                        ;TEST 75        TEST MODE 7 W/ NEGATE
                        ;*******************************************************************************
005574                  TS75:
005574  005000                  CLR     R0              ;R0=0
005576  005010                  CLR     (R0)            ;LOC. 0=0
005600  005110                  COM     (R0)            ;LOC. 0=177777
005602  105100                  COMB    R0              ;R0=377
005604  105470  000005          NEGB    @5(R0)          ;R0+5=404,  404=1,  LOC. 0=777
005610  100403                  BMI     NEG70           ;CC=0001?
```

```
005612  001402                    BEQ     NEG70
005614  102401                    BVS     NEG70
005616  103401                    BCS     NEG71
005620                    NEG70:
005620  104000                    EMT                     ;NEG DID NOT SET CC'S CORRECTLY
005622  105100            NEG71:  COMB    R0              ;R0=0
005624  105120                    COMB    (R0)+           ;LOC. 0=400,  R0=1
005626  105310                    DECB    (R0)            ;LOC. 0=0
005630  005467  172144            NEG     0               ;USE NEG MODE 67 TO TST FOR ZERO
005634  001401                    BEQ     TS76
005636  104000                    EMT                     ;DATA RESULT OF NEG WAS INCORRECT

                          ;*************************************************************************
                          ;
                          ;       THIS TEST VERIFIES PROGRAM COUNTER ADDRESSING WITH SOP
                          ;INSTRUCTIONS.  CLR MODE 77 IS USED TO CLEAR THE LOCATION FOLLOWING THE
                          ;INSTRUCTION (SOPX).  THEN SINGLE OPERAND INSTRUCTIONS WITH MODES 37, 67, AND
                          ;77, USING INDIRECT POINTER SOPXAD ARE USED TO VERIFY THE DATA RESULTS
                          ;OF THESE INSTRUCTIONS.
                          ;
                          ;*************************************************************************
                          ;TEST 76        TEST SOP INSTRUCTIONS MODES 2,3,6,7 WITH REGISTER 7
                          ;*************************************************************************
005640                    TS76:
005640  005027                    CLR     (R7)+           ;CLEAR NEXT LOCATION: (SOPX)
005642  177777            SOPX:   -1                      ;USE MODE 27
005644  001401                    BEQ     SOPA
005646  104000                    EMT                     ;CLR DID NOT SET Z-BIT
005650  005237  005642    SOPA:   INC     @#SOPX          ;INC SOPX W/MODE 37
005654  005467  177762            NEG     SOPX            ;NEGATE SOPX W/MODE 67
005660  100003                    BPL     SOPB
005662  005277  000004            INC     @SOPXAD         ;INC SOPX W/MODE 77
005666  001402                    BEQ     TS77
005670                    SOPB:
005670  104000                    EMT                     ;INC DID NOT SET Z-BIT
005672  005642            SOPXAD: SOPX                    ;INDIRECT ADDRESS OF SOPX

                          ;*************************************************************************
                          ;
                          ;       THIS TEST VERIFIES SINGLE OPERAND NON-MODIFYING INSTRUCTIONS
                          ;USING MODE 0.  R0 IS SET TO ZERO AND THE CONDITION CODES ARE SET
                          ;TO THE COMPLEMENT OF THAT EXPECTED BY THE INSTRUCTION.  A TST INSTRUCTION
                          ;IS EXECUTED AND CONDITIONAL BRANCHES ARE USED TO TEST THE CONDITION
                          ;CODES.
                          ;
                          ;*************************************************************************
                          ;TEST 77        TEST MODE 0 SOP NON-MODIFYING
                          ;*************************************************************************
005674                    TS77:
005674  005000                    CLR     R0              ;INITIALIZE R0=0
005676  000277                    SCC                     ;SET CC=1011
005700  000244                    CLZ
005702  005700                    TST     R0              ;TRY TST W/ MODE 0
005704  102403                    BVS     SNM0A           ;CHECK THAT CC=0100
005706  100402                    BMI     SNM0A
005710  103401                    BCS     SNM0A
```

```
005712  001401                  BEQ     TS100
005714                  SNM0A:
005714  104000                  EMT                     ;CONDITION CODES NOT SET PROPERLY

                        ;*************************************************************************
                        ;
                        ;       THIS TEST VERIFIES SINGLE OPERAND NON-MODIFYING BYTE INSTRUCTIONS WITH MODE 0.
                        ;R0 IS SET TO 377 AND COMPLEMENT OF THE EXPECTED CONDITION CODES
                        ;IS LOADED IN PSW.  A TSTB INSTRUCTION IS EXECUTED AND THE RESULTS
                        ;ARE CHECKED WITH SEVERAL CONDITIONAL BRANCH INSTRUCTIONS.
                        ;       THIS VERIFIES THAT THE PROPER BYTE WAS TESTED.
                        ;
                        ;*************************************************************************
                        ;TEST 100       TEST MODE 0 EVEN BYTE W/ SOP NON-MODIFYING
                        ;*************************************************************************
005716                  TS100:
005716  005000                  CLR     R0              ;INITIALIZE
005720  105100                  COMB    R0              ;R0=377
005722  000277                  SCC                     ;SET CC=0111
005724  000250                  CLN
005726  105700                  TSTB    R0              ;TRY TST EVEN BYTE
005730  102402                  BVS     SNMB0A          ;CHECK CC=1000
005732  101401                  BLOS    SNMB0A
005734  100401                  BMI     TS101
005736                  SNMB0A:
005736  104000                  EMT                     ;CONDITION CODES NOT SET PROPERLY

                        ;*************************************************************************
                        ;
                        ;       THIS TEST VERIFIES SINGLE OPERAND INSTRUCTIONS WITH MODE 1.
                        ;R0 IS USED TO POINT TO AND CLEAR LOC. 0.  THE COMPLEMENT OF THE
                        ;EXPECTED CONDITION CODES ARE LOADED IN THE PSW.  A TST INSTRUCTION
                        ;IS THEN EXECUTED ON LOC. 0 USING R0 AND CONDITIONAL BRANCHES TEST
                        ;THE RESULTS.
                        ;
                        ;*************************************************************************
                        ;TEST 101       TEST MODE 1 SOP NON-MODIFYING
                        ;*************************************************************************
005740                  TS101:
005740  005000                  CLR     R0              ;POINT TO LOC 0
005742  005010                  CLR     (R0)            ;CLEAR LOC 0
005744  000277                  SCC                     ;INITIALIZE
005746  000244                  CLZ                     ;CC=1011
005750  005710                  TST     (R0)            ;TRY TST W/ MODE 1
005752  102403                  BVS     SNM1A           ;CHECK CC=0100
005754  103402                  BCS     SNM1A
005756  100401                  BMI     SNM1A
005760  001401                  BEQ     TS102
005762                  SNM1A:
005762  104000                  EMT                     ;CC'S NOT SET PROPERLY

                        ;*************************************************************************
                        ;
                        ;       THIS TEST SETS LOCATION 0 TO 377 AND THEN USES R0 TO TEST
                        ;THE EVEN BYTE AND THE ODD BYTE USING SOP BYTE INSTRUCTIONS WITH MODE 1.
                        ;AGAIN, CONDITIONAL BRANCHES ARE USED TO VERIFY THE SETTING OF THE
```

```
                                 ;PROPER CONDITION CODE BITS.
                                 ;
                                 ;*************************************************************************
                                 ;TEST 102       TEST MODE 1 BYTE INST. NON-MODIFYING
                                 ;*************************************************************************
 005764                          TS102:
 005764  005000                          CLR     R0              ;POINT TO LOC 0
 005766  005010                          CLR     (R0)            ;CLEAR LOC 0
 005770  105110                          COMB    (R0)            ;COMPLEMENT BYTE 0
 005772  000277                          SCC                     ;SET CC=0111
 005774  000250                          CLN
 005776  105710                          TSTB    (R0)            ;TRY TST ON EVEN BYTE
 006000  102402                          BVS     SNMB1A
 006002  101401                          BLOS    SNMB1A
 006004  100401                          BMI     SNMB1B
 006006                          SNMB1A:
 006006  104000                          EMT                     ;CC'S NOT CORRECT
 006010  005000                  SNMB1B: CLR     R0
 006012  005200                          INC     R0
 006014  000277                          SCC                     ;SET CC=1011
 006016  000244                          CLZ
 006020  105710                          TSTB    (R0)            ;TRY TO TST AN ODD BYTE
 006022  102403                          BVS     SNMB1C          ;CHECK CC=0100
 006024  103402                          BCS     SNMB1C
 006026  100401                          BMI     SNMB1C
 006030  001401                          BEQ     TS103
 006032                          SNMB1C:
 006032  104000                          EMT                     ;CC'S NOT CORRECT

                                 ;*************************************************************************
                                 ;
                                 ;       THIS TEST VERIFIES THE SINGLE-OPERAND NON-MODIFYING INSTRUCTIONS
                                 ;USING MODE 2.  IT USES THE IDENTICAL PROCEDURE EMPLOYED IN THE
                                 ;MODE 1 TESTS.  ADDITIONALLY, THE REGISTER IS CHECKED TO ASSURE THAT
                                 ;IT IS INCREMENTED PROPERLY.
                                 ;
                                 ;*************************************************************************
                                 ;TEST 103       TEST MODE 2 WITH SOP NON-MODIFYING
                                 ;*************************************************************************
 006034                          TS103:
 006034  005000                          CLR     R0              ;INITIALIZE R0=0
 006036  005010                          CLR     (R0)            ;CLEAR LOC 0
 006040  000277                          SCC                     ;SET CC=1011
 006042  000244                          CLZ
 006044  005720                          TST     (R0)+           ;TRY TST W/ MODE 2
 006046  102403                          BVS     SNM2A           ;CHECK CC=0100
 006050  103402                          BCS     SNM2A
 006052  100401                          BMI     SNM2A
 006054  001401                          BEQ     SNM2B
 006056                          SNM2A:
 006056  104000                          EMT                     ;CC'S NOT CORRECT
 006060  005300                  SNM2B:  DEC     R0              ;RESET R0
 006062  005300                          DEC     R0
 006064  001401                          BEQ     TS104
 006066  104000                          EMT                     ;MODE 2 DID NOT INC REQ CORRECTLY
```

```
                         ;*************************************************************************
                         ;
                         ;       THIS TEST VERIFIES MODE 2 SINGLE OPERAND NON-MODIFYING BYTE
                         ;INSTRUCTIONS IT USES RO TO POINT TO LOC. 0.  WITH LOCATION 0
                         ;SET TO 377, THE EVEN AND ODD BYTE IS TESTED WITH TSTB INSTRUCTIONS
                         ;TO VERIFY THE CORRECT CC ARE SET.  THE REGISTER IS CHECKED FOR
                         ;PROPER INCREMENTING.
                         ;
                         ;*************************************************************************
                         ;TEST 104       TEST MODE 2 - BYTE W/ SOP NON-MODIFYING
                         ;*************************************************************************
006070                   TS104:
006070  005000                   CLR     R0              ;CLEAR R0
006072  005010                   CLR     (R0)            ;CLEAR LOC 0
006074  105110                   COMB    (R0)            ;SET LOC 0=377
006076  000277                   SCC                     ;SET CC=0111
006100  000250                   CLN
006102  105720                   TSTB    (R0)+           ;TRY TST OF EVEN BYTE
006104  102402                   BVS     SNMB2A
006106  101401                   BLOS    SNMB2A
006110  100401                   BMI     SNMB2B
006112                   SNMB2A:
006112  104000                   EMT                     ;CC'S NOT SET CORRECTLY
006114  005300           SNMB2B: DEC     R0              ;DECREMENT R0
006116  001401                   BEQ     SNMB2C
006120  104000                   EMT                     ;MODE 2 DID NOT INC REG CORRECTLY
006122  005200           SNMB2C: INC     R0              ;POINT TO ODD BYTE
006124  000277                   SCC                     ;SET CC=1011
006126  000244                   CLZ
006130  105720                   TSTB    (R0)+           ;TRY TST OF ODD BYTE
006132  102403                   BVS     SNMB2D          ;CHECK CC'S=0100
006134  103402                   BCS     SNMB2D
006136  100401                   BMI     SNMB2D
006140  001401                   BEQ     SNMB2E
006142                   SNMB2D:
006142  104000                   EMT                     ;CC'S NOT CORRECT
006144  005300           SNMB2E: DEC     R0
006146  005300                   DEC     R0
006150  001401                   BEQ     TS105
006152  104000                   EMT                     ;R0 DID NOT INCREMENT PROPERLY

                         ;*************************************************************************
                         ;
                         ;       THIS TEST VERIFIES MODE 3 SINGLE OPERAND NON-MODIFYING INSTRUCTIONS.
                         ;A POINTER IN A TABLE AT LOC. 376 IS USED TO TEST LOCATION 0.
                         ;THE CC'S AND THE REGISTER ARE CHECKED FOLLOWING THE
                         ;TST MODE 3 INSTRUCTION.
                         ;
                         ;*************************************************************************
                         ;TEST 105       TEST MODE 3 W/ SOP NON-MODIFYING INSTS
                         ;*************************************************************************
006154                   TS105:
006154  005000                   CLR     R0              ;R0=0
006156  005010                   CLR     (R0)            ;CLEAR LOC 0
006160  105100                   COMB    R0              ;R0=376
006162  005300                   DEC     R0
```

```
006164 000277           SCC                     ;SET CC=1011
006166 000244           CLZ
006170 005730           TST     @(R0)+          ;TRY TST W/ MODE 3
006172 102403           BVS     SNM3A           ;CHECK CC=0100
006174 103402           BCS     SNM3A
006176 100401           BMI     SNM3A
006200 001401           BEQ     SNM3B
006202          SNM3A:
006202 104000           EMT                     ;CC'S NOT CORRECT
006204 005300   SNM3B:  DEC     R0              ;R0=377
006206 105100           COMB    R0              ;R0=0
006210 001401           BEQ     TS106
006212 104000           EMT                     ;MODE 3 DID NOT INC REG CORRECTLY

                ;***********************************************************************
                ;
                ;       THIS TEST VERIFIES SOP NON-MODIFYING BYTE INSTRUCTIONS MODE 3
                ;LOC. 0 IS SET TO 377.  TABLE AT LOC. 402-404 IS USED TO TEST
                ;BYTE 0 AND BYTE 1.  THE REGISTER IS CHECKED FOR PROPER INCREMENTING AND
                ;THE CC'S ARE VERIFIED.
                ;       THE TABLE AT LOC. 402-404 SHOULD CONTAIN 0 AND 1 BEFORE AND
                ;AFTER THE TEST IS RUN.
                ;
                ;***********************************************************************
                ;TEST 106       TEST MODE 3 - BYTES W/ SOP NON-MODIFYING INSTS.
                ;***********************************************************************
006214          TS106:
006214 005000           CLR     R0              ;R0=0
006216 005010           CLR     (R0)            ;CLEAR LOC 0
006220 105110           COMB    (R0)            ;LOC. 0 =377
006222 105100           COMB    R0
006224 005200           INC     R0
006226 005720           TST     (R0)+           ;R0=402
006230 000277           SCC                     ;CC=0111
006232 000250           CLN
006234 105730           TSTB    @(R0)+          ;TRY TST OF EVEN BYTE
006236 102402           BVS     SNMB3A          ;CHECK CC=1000
006240 101401           BLOS    SNMB3A
006242 100401           BMI     SNMB3B
006244          SNMB3A:
006244 104000           EMT                     ;CC'S NOT CORRECT
006246 000277   SNMB3B: SCC                     ;SET CC=1011
006250 000244           CLZ
006252 105730           TSTB    @(R0)+          ;TRY TST OF ODD BYTE
006254 102403           BVS     SNMB3C          ;CHECK CC=0100
006256 103402           BCS     SNMB3C
006260 100401           BMI     SNMB3C
006262 001401           BEQ     SNMB3D
006264          SNMB3C:
006264 104000           EMT                     ;CC'S NOT CORRECT
006266 005720   SNMB3D: TST     (R0)+           ;R0=410
006270 005710           TST     (R0)
006272 100401           BMI     TS107
006274 104000           EMT                     ;TSTB DID NOT INCREMENT R0 CORRETCLY
                ;***********************************************************************
                ;
```

```
                                ;       THIS TEST VERIFIES MODE 4 SOP NON-MODIFYING INSTRUCTIONS.
                                ;LOC. 0 IS SET TO -1 AND THE CC'S ARE SET TO THE COMPLEMENT OF THE
                                ;EXPECTED RESULTS.  RO AND SET TO 2 AND A TST MODE 4 IS EXECUTED.
                                ;THE CC'S ARE CHECKED WITH CONDITIONAL BRANCH INSTRUCTIONS AND THE REGISTER
                                ;IS CHECKED FOR PROPER DECREMENTING.
                                ;
                                ;*************************************************************************
                                ;TEST 107       TEST MODE 4 W/ SOP NON-MODIFYING INSTS
                                ;*************************************************************************
 006276                         TS107:
 006276  005000                         CLR     R0              ;R0=0
 006300  005010                         CLR     (R0)            ;LOC 0=0
 006302  005120                         COM     (R0)+           ;LOC 0=-1
 006304  000277                         SCC                     ;SET CC=1011
 006306  000244                         CLZ
 006310  005740                         TST     -(R0)           ;TRY TST W/ MODE 4
 006312  102402                         BVS     SNM4A           ;CHECK CC=0100
 006314  101401                         BLOS    SNM4A
 006316  100401                         BMI     SNM4B
 006320                         SNM4A:
 006320  104000                         EMT                     ;CC'S NOT CORRECT
 006322  005700                 SNM4B:  TST     R0
 006324  001401                         BEQ     TS110
 006326  104000                         EMT                     ;TST MODE 4 DID NOT DEC R0 CORRECTLY

                                ;*************************************************************************
                                ;
                                ;       THIS TEST VERIFIES MODE 5 SOP NON-MODIFYING INSTRUCTIONS.
                                ;IT USES A POINTER AT LOC. 376 TO TEST LOC. 0.  R0 IS SET
                                ;TO 400, A TST MODE 5 INSTRUCTION IS EXECUTED AND THE CC'S CHECKED.
                                ;R0 IS CHECKED TO INSURE PROPER DECREMENTING.
                                ;
                                ;*************************************************************************
                                ;TEST 110       TEST MODE 5 W/ SOP NON-MODIFYING INSTS
                                ;*************************************************************************
 006330                         TS110:
 006330  005000                         CLR     R0              ;R0=0
 006332  005010                         CLR     (R0)            ;LOC 0=0
 006334  005110                         COM     (R0)            ;LOC 0=-1
 006336  105100                         COMB    R0              ;R0=377
 006340  005200                         INC     R0              ;R0=400
 006342  000277                         SCC                     ;SET CC=0111
 006344  000250                         CLN
 006346  005750                         TST     @-(R0)          ;TRY TST W/ MODE 5
 006350  102402                         BVS     SNM5A           ;CHECK CC=1000
 006352  101401                         BLOS    SNM5A
 006354  100401                         BMI     SNM5B
 006356                         SNM5A:
 006356  104000                         EMT                     ;CC'S NOT SET PROPERLY
 006360  005200                 SNM5B:  INC     R0              ;R0=377
 006362  105100                         COMB    R0              ;R0=0
 006364  001401                         BEQ     TS111
 006366  104000                         EMT                     ;MODE 5 DID NOT DEC R0 CORRECTLY

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

```
                                ;       THIS TEST VERIFIES MODE 6 SOP NON-MODIFYING INSTRUCTIONS.
                                ;R0 IS SET TO 377 AND A MODE 6 TST INSTRUCTION IS EXECUTED
                                ;USING R0 AND AN OFFSET OF -377.  THE CC'S ARE CHECKED AS WELL
                                ;AS R0 TO INSURE IT WAS NOT ALTERED.
                                ;
                                ;**********************************************************************
                                ;TEST 111       TEST MODE 6 W/ SOP NON-MODIFYING INSTS
                                ;**********************************************************************
006370                          TS111:
006370  005000                          CLR     R0              ;R0=0
006372  005010                          CLR     (R0)            ;LOC 0=0
006374  005110                          COM     (R0)            ;LOC 0=-1
006376  105100                          COMB    R0              ;R0=377
006400  000277                          SCC                     ;SET CC=0111
006402  000250                          CLN
006404  005760  177401                  TST     -377(R0)        ;TRY TST W/ MODE 6
006410  102402                          BVS     SNM6A           ;CHECK CC=1000
006412  101401                          BLOS    SNM6A
006414  100401                          BMI     SNM6B
006416                          SNM6A:
006416  104000                          EMT                     ;CC'S INCORRECT
006420  105100                  SNM6B:  COMB    R0              ;R0=0
006422  001401                          BEQ     TS112
006424  104000                          EMT                     ;TST MODE 6 INCORRECTLY CHANGED R0
```

```

                                        ;*******************************************************************************
                                        ;
                                        ;       THIS TEST VERIFIES MODE 7 SOP NON-MODIFYING INSTRUCTIONS.
                                        ;IT USES A POINTER TO LOC. 0 STORED AT LOC. 400 TO TST LOC. 0.
                                        ;R0 IS SET TO 377 AND LOC. 0 IS TESTED THRU THE POINTER AT 400 USING
                                        ;R0 AND AN OFFSET OF 1.
                                        ;
                                        ;*******************************************************************************
                                        ;TEST 112       TEST MODE 7 W/ SOP NON-MODIFYING INSTS.
                                        ;*******************************************************************************
 006426                                 TS112:
 006426  005000                                 CLR     R0              ;R0=0
 006430  005010                                 CLR     (R0)            ;LOC 0=0
 006432  005110                                 COM     (R0)            ;LOC 0=-1
 006434  105100                                 COMB    R0              ;R0=377
 006436  000277                                 SCC                     ;CC=0111
 006440  000250                                 CLN
 006442  005770  000001                         TST     @1(R0)          ;TRY TST W/ MODE 7
 006446  102402                                 BVS     SNM7A           ;CHECK CC=1000
 006450  101401                                 BLOS    SNM7A
 006452  100401                                 BMI     SNM7B
 006454                                 SNM7A:
 006454  104000                                 EMT                     ;CC'S NOT CORRECT
 006456  105100                         SNM7B:  COMB    R0              ;R0=0
 006460  001401                                 BEQ     TS113
 006462  104000                                 EMT                     ;TST MODE 7 INCORRECTLY CHANGED R0

                                        ;*******************************************************************************
                                        ;
                                        ;       THIS TEST VERIFIES MODE 0 DOUBLE OPERAND INSTRUCTIONS.  IT SETS
                                        ;DATA IN R0 AND R4 AND USES THE ADD INSTRUCTION TO TEST THE DOP
                                        ;MICROCODE.
                                        ;
                                        ;*******************************************************************************
                                        ;TEST 113       TEST MODE 0 DOUBLE-OPERAND (DOP) INSTS.
                                        ;*******************************************************************************
 006464                                 TS113:
 006464  005000                                 CLR     R0              ;R0=0
 006466  005100                                 COM     R0              ;R0=-1
 006470  005004                                 CLR     R4              ;R4=0
 006472  060004                                 ADD     R0,R4           ;TRY ADD: R4=-1
 006474  005204                                 INC     R4              ;R4=0
 006476  001401                                 BEQ     TS114
 006500  104000                                 EMT                     ;ADD INST. FAILED W/ MODE 0

                                        ;*******************************************************************************
                                        ;
                                        ;       THIS TEST VERIFIES THE MOVE INSTRUCTION WITH MODE 0 TO MODE 0.
                                        ;
                                        ;*******************************************************************************
                                        ;TEST 114       MOV MODE 0 TO MODE 0
                                        ;*******************************************************************************
 006502                                 TS114:
 006502  005000                                 CLR     R0              ;R0=0
 006504  005004                                 CLR     R4              ;R4=0
```

```
006506  005100                  COM     R0              ;R0=-1
006510  010004                  MOV     R0,R4           ;TRY MOVE -1 TO R4
006512  005204                  INC     R4              ;INC R4
006514  001401                  BEQ     TS115
006516  104000                  EMT                     ;MOVE FAILED MODE 0 TO MODE 0

                        ;*************************************************************************
                        ;
                        ;       THIS TEST VERIFIES THE SUBTRACT INSTRUCTION WITH MODE 0,0.
                        ;
                        ;*************************************************************************
                        ;TEST 115       TEST SUB MODE 0,0
                        ;*************************************************************************
006520                  TS115:
006520  005000                  CLR     R0              ;R0=0
006522  005004                  CLR     R4              ;R4=0
006524  005204                  INC     R4              ;R4=1
006526  160400                  SUB     R4,R0           ;TRY SUB 0,0   R0=-1
006530  100003                  BPL     SUB0            ;CC=1001
006532  001402                  BEQ     SUB0
006534  102401                  BVS     SUB0
006536  103401                  BCS     SUB0A
006540                  SUB0:
006540  104000                  EMT                     ;CONDITION CODE FAILED ON SUB
006542  005200          SUB0A:  INC     R0
006544  001401                  BEQ     TS116
006546  104000                  EMT                     ;DATA RESULT OF SUB FAILED

                        ;*************************************************************************
                        ;
                        ;       THIS TEST QUICKLY VERIFIES THE REMAINING DOP MODIFYING INSTRUCTIONS
                        ;WITH MODE 0,0 TO PROVIDE A BASELINE FOR SUBSEQUENT TESTS.
                        ;SINGLE OPERAND INSTRUCTIONS ARE USED TO SET UP DATA IN R0 AND R4
                        ;BEFORE EACH OF THE SEVERAL DOP MODIFYING INSTRUCTIONS ARE USED AND
                        ;VERIFIED.
                        ;
                        ;*************************************************************************
                        ;TEST 116       TEST ALL THE DOP INSTRUCTIONS W/ SOURCE MODE 0,0
                        ;*************************************************************************
006550                  TS116:
006550  005000                  CLR     R0              ;R0=0
006552  010004                  MOV     R0,R4           ;TRY MOVE MODE 0,0
006554  001401                  BEQ     DOP0A
006556  104000                  EMT                     ;Z-BIT NOT SET
006560  005200          DOP0A:  INC     R0              ;R0=1
006562  005100                  COM     R0              ;R0=177776
006564  005104                  COM     R4              ;R4=177777
006566  040004                  BIC     R0,R4           ;TRY BIC: R4=1
006570  005304                  DEC     R4              ;R4=0
006572  001401                  BEQ     DOP0B
006574  104000                  EMT                     ;BIC CLEAR RESULT INCORRECT
006576  050004          DOP0B:  BIS     R0,R4           ;TRY BIS: R4=177777
006600  005204                  INC     R4
006602  005204                  INC     R4              ;R4=0
006604  001401                  BEQ     DOP0C
006606  104000                  EMT                     ;RESULT OF BIS INCORRECT
```

```
006610  005000                  DOPOC:  CLR     R0              ;R0=0
006612  105100                          COMB    R0              ;R0=377
006614  005004                          CLR     R4              ;R4=0
006616  005104                          COM     R4              ;R4=177777
006620  040004                          BIC     R0,R4           ;R4=177400
006622  060004                          ADD     R0,R4           ;TRY ADD: R4=177777
006624  005204                          INC     R4              ;R4=0
006626  001401                          BEQ     DOPOD
006630  104000                          EMT                     ;RESULT OF ADD INCORRECT
006632  160004                  DOPOD:  SUB     R0,R4           ;177401=R4
006634  105404                          NEGB    R4              ;R4=177777
006636  005204                          INC     R4              ;RD=0
006640  001401                          BEQ     TS117
006642  104000                          EMT                     ;RESULT OF SUB INCORRECT

                                ;*******************************************************************************
                                ;
                                ;       THIS TEST VERIFIES MODE 0,X DOUBLE OPERAND INSTRUCTIONS.  IT SETS
                                ;DATA IN R0 AND LOCATION 0 AND OPERATES UPON IT USING DOP INSTRUCTIONS.
                                ;
                                ;*******************************************************************************
                                ;TEST 117       TEST MODE 0,X DOUBLE-OPERAND INSTRUCTIONS
                                ;*******************************************************************************
006644                          TS117:
006644  005000                          CLR     R0              ;R0=0
006646  005010                          CLR     (R0)            ;LOC. 0=0
006650  105110                          COMB    (R0)            ;LOC. 0=377
006652  005220                          INC     (R0)+           ;LOC. 0=400   R0=2
006654  005400                          NEG     R0              ;R0=-2
006656  060037  000000                  ADD     R0,@#0          ;TRY ADD 0,3;  LOC. 0=376
006662  100403                          BMI     DOP03A          ;CC=0001?
006664  001402                          BEQ     DOP03A
006666  102401                          BVS     DOP03A
006670  103401                          BCS     DOP03B
006672                          DOP03A:
006672  104000                          EMT                     ;CC'S NOT SET CORRECTLY
006674  105137  000000          DOP03B: COMB    @#0             ;LOC. 0=1
006700  005337  000000                  DEC     @#0             ;LOC. 0=0
006704  001401                          BEQ     TS120
006706  104000                          EMT                     ;DATA RESULT INCORRECT
                                ;*******************************************************************************
                                ;
                                ;       THIS TEST VERIFIES MODE 0,0 DOP NON-MODIFYING INSTRUCTIONS.
                                ;R0 AND R4 ARE PRESET TO 0 AND 1 RESPECTIVELY.  COMPARE INSTRUCTIONS ARE
                                ;THEN EXECUTED AND CHECKED.  FIRST R4 IS COMPARED TO R0 THEN R0 TO R4.
                                ;
                                ;*******************************************************************************
                                ;TEST 120       TEST DOP NON-MODIFYING INST. W/ SOURCE MODE 0,0
                                ;*******************************************************************************
006710                          TS120:
006710  005000                          CLR     R0              ;R0=0
006712  005004                          CLR     R4              ;R4=0
006714  005204                          INC     R4              ;R4=1
006716  020400                          CMP     R4,R0           ;TRY COMPARE R4 TO R0
006720  003001                          BGT     DNM1
006722  104000                          EMT                     ;CC'S NOT CORRECT FOR CMP
```

```
006724  020004                  DNM1:   CMP     R0,R4           ;TRY COMPARE R0 TO R4
006726  002401                          BLT     DNM2
006730  104000                          EMT                     ;CC'S NOT CORRECT FOR CMP
006732  005200                  DNM2:   INC     R0              ;R0=1
006734  020400                          CMP     R4,R0           ;TRY COMPARE R4=1 TO R0=1
006736  001401                          BEQ     DNM3
006740  104000                          EMT                     ;CC'S NOT CORRECT (Z=1) FOR CMP
006742  005000                  DNM3:   CLR     R0              ;R0=0
006744  005100                          COM     R0              ;R0=177777
006746  005004                          CLR     R4              ;R4=0
006750  030004                          BIT     R0,R4           ;TRY BIT R0 TO R4
006752  001401                          BEQ     DNM4
006754  104000                          EMT                     ;CC'S NOT CORRECT FOR BIT
006756  005304                  DNM4:   DEC     R4              ;R4=177777
006760  030004                          BIT     R0,R4           ;TRY BIT AGAIN
006762  100401                          BMI     TS121
006764  104000                          EMT                     ;CC'S NOT CORRECT FOR BIT
                                ;*********************************************************************************
                                ;
                                ;       THIS TEST VERIFIES MODE 0,X DOUBLE OPERAND NON-MODIFYING INSTRUCTIONS.
                                ;IT SETS DATA IN R0 AND LOCATION 0 AND COMPARES THEM USING DOPNM INSTRUCTIONS.
                                ;
                                ;*********************************************************************************
                                ;TEST 121       TEST MODE 0,X DOUBLE-OPERAND NON-MODIFYING INSTS.
                                ;*********************************************************************************
006766                          TS121:
006766  005000                          CLR     R0              ;R0=0
006770  005010                          CLR     (R0)            ;LOC. 0=0
006772  005110                          COM     (R0)            ;LOC. 0=177777
006774  005200                          INC     R0              ;R0=1
006776  020037  000000                  CMP     R0,@#0          ;TRY CMP MODE 0,3
007002  100403                          BMI     DNM03A          ;CC=0001
007004  001402                          BEQ     DNM03A
007006  102401                          BVS     DNM03A
007010  103401                          BCS     DNM03B
007012                          DNM03A:
007012  104000                          EMT                     ;CC'S NOT SET CORRECTLY
007014  005300                  DNM03B: DEC     R0
007016  001002                          BNE     DNM03C
007020  005210                          INC     (R0)
007022  001401                          BEQ     TS122
007024                          DNM03C:
007024  104000                          EMT                     ;DATA INCORRECTLY MODIFIED BY CMP
                                ;************* *******************************************************************
                                ;
                                ;       THIS TEST VERIFIES MODE 1 DOP INSTRUCTIONS.  R0 IS SET TO -1
                                ;AND LOC 0 TO 1.  R4 IS THEN CLEARED AND USED TO POINT TO LOC 0.
                                ;IN THE ADD MODE 1 INSTRUCTION, LOC 0 IS ADDED TO R0 AND THE
                                ;RESULTS VERIFIED.
                                ;
                                ;*********************************************************************************
                                ;TEST 122       TEST MODE 1 W/ DOP INST.
                                ;*********************************************************************************
007026                          TS122:
007026  005000                          CLR     R0              ;R0=0
007030  005100                          COM     R0              ;R0=177777
```

```
007032  005004          CLR     R4              ;R4=0
007034  005014          CLR     (R4)            ;LOC 0=0
007036  005214          INC     (R4)            ;LOC 0=1
007040  061400          ADD     (R4),R0         ;TRY ADD SOURCE MODE 1
007042  001401          BEQ     TS123
007044  104000          EMT                     ;RESULT OF ADD INCORRECT

                ;*******************************************************************************
                ;
                ;       THIS TEST VERIFIES MODE 1 DOP BYTE INSTRUCTIONS WHICH ADDRESS
                ;EVEN BYTES.  LOC. 0 IS SET TO -1 AND R4 IS CLEARED.  THEN R4 IS
                ;SET TO -1 USING A BISB THRU R0 WITH MODE 1.
                ;
                ;*******************************************************************************
                ;TEST 123       TEST MODE 1 - EVEN BYTE W/ DOP INSTS.
                ;*******************************************************************************
007046          TS123:
007046  005000          CLR     R0              ;R0=0
007050  005010          CLR     (R0)            ;LOC. 0=0
007052  005110          COM     (R0)            ;LOC. 0=177777
007054  005004          CLR     R4              ;R4=0
007056  151004          BISB    (R0),R4         ;TRY MODE 1- EVEN BYTE W/ DOP
007060  105104          COMB    R4              ;R4=0
007062  001401          BEQ     TS124
007064  104000          EMT                     ;RESULT OF BISB IS INCORRECT

                ;*******************************************************************************
                ;
                ;       THIS TEST VERIFIES MODE 1 DOP NON-MODIFYING INSTRUCTIONS
                ;WHICH ADDRESS EVEN BYTES.  LOC. 0 IS SET TO -1 AND R0 IS CLEARED
                ;AND USED AS THE ADDRESSING REGISTER.  R4 IS SET TO 377 AND A
                ;MODE 1,0 CMPB INSTRUCTION IS USED THE RESULTS VERIFIED.
                ;
                ;*******************************************************************************
                ;TEST 124       TEST MODE 1 - EVEN BYTE W/ DOP NON-MODIFYING INST.
                ;*******************************************************************************
007066          TS124:
007066  005000          CLR     R0              ;R0=0
007070  005010          CLR     (R0)            ;LOC 0=0
007072  005110          COM     (R0)            ;LOC 0=177777
007074  005004          CLR     R4              ;R4=0
007076  105104          COMB    R4              ;R4=377
007100  121004          CMPB    (R0),R4         ;TRY MODE 1 - EVEN BYTE W/ DOP NON-MODIFYING
007102  001401          BEQ     TS125
007104  104000          EMT                     ;RESULT OF CMPB INCORRECT

                ;*******************************************************************************
                ;
                ;       THIS TEST VERIFIES MODE 1,0 MOVB INSTRUCTIONS
                ;WHICH ADDRESS EVEN BYTES.  LOC. 0 IS SET TO 177400, R0 IS CLEARED AND
                ;R4 IS SET TO -1.  MOVB ARE USED TO MOVE BYTE 0 TO R4.  THIS
                ;VERIFIES THAT THE PROPER BYTE WAS SELECTED AND THAT THE SIGN-X-TEND
                ;FUNCTION WITH MODE 0.
                ;       THEN LOC. 0 IS COMPLEMENTED AND THE SAME PROCEDURE EXERCISES
                ;THE LOGIC FOR COMPLEMENTARY DATA.
                ;       THIS TEST EXERCISES UNIQUE MICROCODE.
```

```
                                ;
                                ;*****************************************************************************
                                ;TEST 125       TEST MOV INSTRUCTION MODE 1,0 EVEN BYTE
                                ;*****************************************************************************
007106                          TS125:
007106  005000                          CLR     R0                      ;R0=0
007110  005010                          CLR     (R0)                    ;LOC 0=0
007112  105110                          COMB    (R0)                    ;LOC 0=177400
007114  005110                          COM     (R0)
007116  005004                          CLR     R4                      ;R4=0
007120  005104                          COM     R4                      ;R4=177777
007122  111004                          MOVB    (R0),R4                 ;R4=0
007124  005704                          TST     R4                      ;CHECK SIGN OF WORD
007126  001401                          BEQ     DOP1
007130  104000                          EMT                             ;MOVB SHOULD SIGN X-TEND
007132  005110                  DOP1:   COM     (R0)                    ;LOC 0=177777
007134  111004                          MOVB    (R0),R4                 ;DO MOVB W/ EVEN BYTE
007136  100401                          BMI     TS126
007140  104000                          EMT                             ;MOVB SHOULD SIGN X-TEND

                                ;*****************************************************************************
                                ;
                                ;       THIS TEST VERIFIES MODE 1 DOP INSTRUCTIONS WHICH REFERENCE
                                ;ODD BYTES.  LOC. 0 IS SET TO 177400.  R0 IS SET TO 0 AND R4 IS
                                ;SET TO 1.  THE BISB INSTRUCTION USES THE DATA IN BYTE 1 TO SET BYTE 0.
                                ;THE RESULT IS CHECKED BY INCREMENTING THE WORD (LOC. 0) TO ZERO.
                                ;
                                ;*****************************************************************************
                                ;TEST 126       TEST MODE 1-ODD BYTE W/ DOP INSTS.
                                ;*****************************************************************************
007142                          TS126:
007142  005000                          CLR     R0                      ;R0=0
007144  005010                          CLR     (R0)                    ;LOC. 0=0
007146  005004                          CLR     R4                      ;R4=0
007150  005204                          INC     R4                      ;R4=1
007152  105114                          COMB    (R4)                    ;LOC. 0=177400
007154  151410                          BISB    (R4),(R0)               ;TRY TO BIS LOW ORDER BITS W/ MODE 1
007156  005210                          INC     (R0)                    ;CHECK RESULT
007160  001401                          BEQ     TS127
007162  104000                          EMT                             ;RESULT OF BISB INCORRECT

                                ;*****************************************************************************
                                ;
                                ;       THIS TEST VERIFIES MODE 2 DOP INSTRUCTIONS.  LOC. 0 IS SET TO -1.
                                ;R0 IS CLEARED AND USED AS THE MODE 2 ADDRESSING REGISTER TO MOVE LOC. 0
                                ;TO R7.  THE DATA RESULTS ARE VERIFIED AND THE INCREMENTING OF THE REGISTER
                                ;IS CHECKED.
                                ;
                                ;*****************************************************************************
                                ;TEST 127       TEST MODE 2 W/ DOP INSTS.
                                ;*****************************************************************************
007164                          TS127:
007164  005000                          CLR     R0                      ;R0=0
007166  005010                          CLR     (R0)                    ;LOC. 0=0
007170  005110                          COM     (R0)                    ;LOC. 0=177777
007172  012004                          MOV     (R0)+,R4                ;TRY MOVE MODE 2,0
```

```
007174 005204                    INC     R4              ;CHECK R4
007176 001401                    BEQ     DOP2
007200 104000                    EMT                     ;RESULT OF MOV INST INCORRECT
007202 005300           DOP2:    DEC     R0              ;TEST R0 AFTER MODE 2
007204 005300                    DEC     R0
007206 001401                    BEQ     TS130
007210 104000                    EMT                     ;REGISTER NOT INCREMENTED IN MODE 2

                        ;**************************************************************************
                        ;
                        ;        THIS TEST VERIFIES MODE 2 DOP BYTE INSTRUCTIONS WHICH ADDRESS
                        ;EVEN BYTES.  LOC. 0 IS SET TO -1.  R0 IS CLEARED AND USED AS THE
                        ;ADDRESSING REGISTER IN A TEST WHICH TRIES TO CLEAR BYTE 1 USING
                        ;BYTE 0 DATA AND A BICB.  UNIQUE IN THIS TEST IS USE OF THE
                        ;SAME ADDRESSING REGISTER FOR BOTH SOURCE AND DESTINATION.  THE SOURCE AND
                        ;DESTINATION IS CHECKED TO INSURE PROPER FUNCTIONING.
                        ;
                        ;**************************************************************************
                        ;TEST 130        TEST MODE 2 - EVEN BYTE W/ DOP INST.
                        ;**************************************************************************
007212                  TS130:
007212 005000                    CLR     R0              ;R0=0
007214 010010                    MOV     R0,(R0)         ;LOC. 0=0
007216 005110                    COM     (R0)            ;LOC. 0=177777
007220 142010                    BICB    (R0)+,(R0)      ;TRY TO CLEAR BYTE 1 FROM BYTE 0 W/ BICB
007222 105737 000001             TSTB    @#1             ;CHECK RESULT
007226 001401                    BEQ     DOPB2A
007230 104000                    EMT                     ;BICB DESTINATION INCORRECT
007232 105137 000000    DOPB2A:  COMB    @#0             ;CHECK BICB SOURCE
007236 001401                    BEQ     TS131
007240 104000                    EMT                     ;BICB SOURCE INCORRECTLY CHANGED
                        ;**************************************************************************
                        ;
                        ;        THIS TEST VERIFIES MODE 2 DOP BYTE INSTRUCTIONS WHICH REFERENCE
                        ;ODD BYTES.  R0 IS SET TO 1, LOC. 0 IS SET TO 177400, AND R4 IS CLEARED.
                        ;A MODE 2 MOVB USES R0 TO MOVE BYTE 1 TO R4.  AN INCREMENT
                        ;IS USED TO CHECK THAT THE PROPER BYTE WAS MOVED AND SIGN X-TENDED
                        ;
                        ;**************************************************************************
                        ;TEST 131        TEST MODE 2 - ODD BYTE W/ DOP INST.
                        ;**************************************************************************
007242                  TS131:
007242 005000                    CLR     R0              ;R0=0
007244 005004                    CLR     R4              ;R4=0
007246 005010                    CLR     (R0)            ;LOC. 0=0
007250 005110                    COM     (R0)            ;LOC. 0=177777
007252 105120                    COMB    (R0)+           ;LOC 0=177400; R0=1
007254 112004                    MOVB    (R0)+,R4        ;TRY DOP MODE 2 W/ ODD BYTE
007256 005204                    INC     R4              ;CHECK RESULT OF MOVB
007260 001401                    BEQ     DOPB2B
007262 104000                    EMT                     ;RESULT OF MOVB INCORRECT
007264 005740           DOPB2B:  TST     -(R0)           ;BUMP R0 DOWN BY 2
007266 005700                    TST     R0              ;CHECK R0
007270 001401                    BEQ     TS132
007272 104000                    EMT                     ;MODE 2 BYTE DID NOT INCREMENT REG. CORRECTLY
                        ;**************************************************************************
```

```
                                           ;
                                           ;       THIS TEST VERIFIES MODE 3 DOUBLE-OPERAND INSTRUCTIONS.
                                           ;LOC. 0 IS LOADED WITH ALTERNATING ZEROES AND ONES; AND R0 IS LOADED
                                           ;WITH ALTERNATING ONES AND ZEROES.  A MODE 3 BIS IS USED TO SET R0
                                           ;TO -1 BY USING LOC. 0 AS THE SOURCE TO BIS THE ZEROES IN R0.  THE
                                           ;RESULT IS TESTED BY INCREMENTING R0 AND CHECKING FOR ZERO.
                                           ;
                                           ;*************************************************************************
                                           ;TEST 132       TEST MODE 3 W/ DOP INSTS.
                                           ;*************************************************************************
007274                                     TS132:
007274  012737  052525  000000                     MOV     #052525,@#0       ;MOVE 52525 TO LOC. 0
007302  012700  125252                             MOV     #125252,R0        ;SET ALT. ONE AND ZERO IN R0
007306  053700  000000                             BIS     @#0,R0            ;TRY TO SET ALL OTHER BITS W/ MODE 3
007312  005200                                     INC     R0                ;TEST RESULT
007314  001401                                     BEQ     TS133
007316  104000                                     EMT                       ;BIS W/ MODE 3 INCORRECT RESULT
                                           ;*************************************************************************
                                           ;
                                           ;       THIS TEST VERIFIES MODE 3 DOUBLE OPERAND BYTE INSTRUCTIONS WHICH
                                           ;ADDRESS EVEN BYTES.  BYTE 0 IS SET TO ALTERNATING 1'S AND 0'S; BYTE 1,
                                           ;ALTERNATING 0'S AND 1'S.  R0 IS CLEARED AND A BISB IS USED TO
                                           ;SET THE LOW BYTE OF R0 TO 252.
                                           ;
                                           ;*************************************************************************
                                           ;TEST 133       TEST MODE 3 - EVEN BYTE W/ DOP INSTS.
                                           ;*************************************************************************
007320                                     TS133:
007320  012737  052652  000000                     MOV     #52652,@#0        ;MOVE 1'S AND 0' PATTERN TO LOC. 0
007326  005000                                     CLR     R0                ;R0=0
007330  153700  000000                             BISB    @#0,R0            ;TRY R0=252 W/ MODE 3 - EVEN BYTE
007334  022700  000252                             CMP     #252,R0           ;BISB W/ EVEN BYTE SUCCESSFUL?
007340  001401                                     BEQ     TS134
007342  104000                                     EMT                       ;BISB W/ MODE 3 - EVEN BYTE FAILED
                                           ;*************************************************************************
                                           ;
                                           ;       THIS TEST VERIFIES MODE 3 DOUBLE OPERAND BYTE INSTRUCTIONS
                                           ;WHICH ADDRESS ODD BYTES.  THE SAME PROCEDURE USED IN PREVIOUS
                                           ;TEST IS USED HERE.  THIS TIME BYTE 1 IS USED AS THE SOURCE BYTE.
                                           ;THE EXPECTED RESULT IS:  R0 = 125.
                                           ;
                                           ;*************************************************************************
                                           ;TEST 134       TEST MODE 3 - ODD BYTE W/ DOP INSTS.
                                           ;*************************************************************************
007344                                     TS134:
007344  012737  052652  000000                     MOV     #52652,@#0        ;MOVE 1'S AND 0'S PATTERN TO LOC 0
007352  005000                                     CLR     R0                ;R0=0
007354  153700  000001                             BISB    @#1,R0            ;TRY R0=152 W/ MODE 3 - ODD BYTE
007360  022700  000125                             CMP     #125,R0           ;R0=125?
007364  001401                                     BEQ     TS135
007366  104000                                     EMT                       ;BISB W/ MODE 3 - ODD BYTE FAILED

                                           ;*************************************************************************
                                           ;TEST 135       TEST DEST. MODE 0-BYTE W/ DOP NON-MODIFYING MST
                                           ;*************************************************************************
007370                                     TS135:
```

```
007370  005000                   CLR     R0              ;R0=0
007372  105100                   COMB    R0              ;R0=377
007374  000263                   +SEC!SEV                ;SET C AND V BITS
007376  132700  000200           BITB    #200,R0         ;TRY DOPNM DEST. MODE 0-BYTE
007402  001403                   BEQ     DNMB0A          ;BR TO ERROR IF Z BIT SET
007404  102402                   BVS     DNMB0A          ;BR TO ERROR IF V BIT SET
007406  103001                   BCC     DNMB0A          ;BR TO ERROR IF C BIT CLEAR.
007410  100401                   BMI     DNMB0B
007412                   DNMB0A:
007412  104000                   EMT                     ;CC'S INCORRECT
007414  105100           DNMB0B: COMB    R0              ;CHECK DESTINATION DATA
007416  001401                   BEQ     TS136
007420  104000                   EMT                     ;DEST. DATA MODIFIED

                         ;*********************************************************************************
                         ;TEST 136        TEST DEST. MODE 1 W/ DOP NON-MODIFYING INST
                         ;*********************************************************************************
007422                   TS136:
007422  005000                   CLR     R0              ;R0=0
007424  005010                   CLR     (R0)            ;LOC. 0=0
007426  000241                   CLC                     ;CLEAR C BIT
007430  032710  177777           BIT     #177777,(R0)    ;TRY DOPNM DEST. MODE 1
007434  100403                   BMI     DNM1A           ;BR TO ERROR IF N BIT SET
007436  102402                   BVS     DNM1A           ;BR TO ERROR IF V BIT SET
007440  103401                   BCS     DNM1A           ;BR TO ERROR IF C BIT SET
007442  001401                   BEQ     DNM1B
007444                   DNM1A:
007444  104000                   EMT                     ;COND. CODES INCORRECT
007446  005710           DNM1B:  TST     (R0)            ;CHECK TEST DATA
007450  001401                   BEQ     TS137
007452  104000                   EMT                     ;DESTINATION DATA MODIFIED

                         ;*********************************************************************************
                         ;TEST 137        TEST DEST. MODE 2 W/ DOP NON-MODIFYING INST.
                         ;*********************************************************************************
007454                   TS137:
007454  005000                   CLR     R0              ;R0=0
007456  005010                   CLR     (R0)            ;LOC. 0=0
007460  052710  125252           BIS     #125252,(R0)    ;LOC. 0=125252
007464  032720  077777           BIT     #77777,(R0)+    ;TRY DOPNM INST W/ MODE 2
007470  102402                   BVS     DNM2A           ;BR TO ERROR IF V BIT SET
007472  001401                   BEQ     DNM2A           ;BR TO ERROR IF Z-BIT SET
007474  100001                   BPL     DNM2B
007476                   DNM2A:
007476  104000                   EMT                     ;COND. CODES INCORRECT
007500  005300           DNM2B:  DEC     R0              ;DECREMENT R0 TO CHECK IT.
007502  005300                   DEC     R0
007504  001401                   BEQ     DNM2D
007506                   DNM2C:
007506  104000                   EMT                     ;MODE 2 REGISTER NOT INCREMENTED BY 2
007510  022710  125252   DNM2D:  CMP     #125252,(R0)    ;CHECK DEST. DATA
007514  001401                   BEQ     TS140
007516  104000                   EMT                     ;DEST. DATA MODIFIED

                         ;*********************************************************************************
                         ;TEST 140        TEST DEST. MODE 2-BYTE, W/DOP NON-MODIFYING INST
```

```
                                        ;*********************************************************************************
        007520                          TS140:
        007520  005000                          CLR     R0                      ;R0=0
        007522  005010                          CLR     (R0)                    ;LOC. 0=0
        007524  052710  052652                  BIS     #52652,(R0)             ;LOC. 0=52652
        007530  000263                          +SEC!SEV                        ;SET C AND V BITS
        007532  132720  000201                  BITB    #201,(R0)+              ;TRY DOPNM INST. W/ MODE 2 EVEN BYTE
        007536  001403                          BEQ     DNMB2A                  ;BR TO ERROR IF Z-BIT SET
        007540  103002                          BCC     DNMB2A                  ;BR TO ERROR IF C-BIT CLEAR
        007542  102401                          BVS     DNMB2A                  ;BR TO ERROR IF V-BIT SET
        007544  100401                          BMI     DNMB2B
        007546                          DNMB2A:
        007546  104000                          EMT                             ;COND. CODES INCORRECT
        007550  005300                  DNMB2B: DEC     R0                      ;CHECK DEST. REGISTER.
        007552  001401                          BEQ     DNMB2C
        007554  104000                          EMT                             ;DEST. REGISTER NOT INCREMENTED BY 1
        007556  005200                  DNMB2C: INC     R0                      ;R0=1
        007560  132720  000201                  BITB    #201,(R0)+              ;TRY DOPNM INST. W/MODE 2-ODD BYTE
        007564  001402                          BEQ     DNMB2D  ;BR TO ERROR IF Z-BIT SET
        007566  102401                          BVS     DNMB2D                  ;BR TO ERROR IF V-BIT SET
        007570  100001                          BPL     DNMB2E
        007572                          DNMB2D:
        007572  104000                          EMT                             ;COND. CODES INCORRECT
        007574  005300                  DNMB2E: DEC     R0                      ;DEC R0 TO CHECK IT.
        007576  005300                          DEC     R0
        007600  001401                          BEQ     DNMB2F
        007602  104000                          EMT                             ;DEST. REGISTER NOT INCREMENTED BY 1
        007604  022710  052652          DNMB2F: CMP     #52652,(R0)             ;CHECK DEST. DATA IS UNMODIFIED
        007610  001401                          BEQ     TS141
        007612  104000                          EMT                             ;DEST. DATA WAS MODIFIED.


                                        ;*********************************************************************************
                                        ;TEST 141       TEST DEST. MODE 3-BYTES W/DOP NON-MODIFYING INST.
                                        ;*********************************************************************************
        007614                          TS141:
        007614  005000                          CLR     R0                      ;R0=0
        007616  005010                          CLR     (R0)                    ;LOC. 0=0
        007620  052710  125125                  BIS     #125125,(R0)            ;LOC. 0=125125
        007624  105100                          COMB    R0                      ;R0=377
        007626  005200                          INC     R0                      ;R0=400
        007630  005010                          CLR     (R0)                    ;LOC. 400=0
        007632  000263                          +SEC!SEV                        ;C-BIT=V-BIT=1
        007634  132730  000201                  BITB    #201,@(R0)+             ;TRY DOPNM W/MODE 3-EVEN BYTE
        007640  001403                          BEQ     DNMB3A                  ;BR TO ERROR IF Z BIT SET
        007642  102402                          BVS     DNMB3A                  ;BR TO ERROR IF V BIT SET
        007644  103001                          BCC     DNMB3A                  ;BR TO ERROR IF C BIT CLEAR
        007646  100001                          BPL     DNMB3B
        007650                          DNMB3A:
        007650  104000                          EMT                             ;COND. CODES INCORRECT
        007652  022700  000402          DNMB3B: CMP     #402,R0                 ;CHECK DEST. REGISTER INC. BY 2 AND INC BY 2 AGAIN
        007656  001401                          BEQ     DNMB3C
        007660  104000                          EMT                             ;DEST. REGISTER NOT INCREMENTED BY 2
        007662  005200                  DNMB3C: INC     R0                      ;R0=404
        007664  005200                          INC     R0
        007666  132730  000201                  BITB    #201,@(R0)+             ;TRY DOPNM DEST MODE 3-BYTE(ODD)
```

```
007672  001402                          BEQ     DNMB3D          ;BR TO ERROR IF Z BIT SET
007674  102401                          BVS     DNMB3D          ;BR TO ERROR IF V BIT SET
007676  100401                          BMI     DNMB3E
007700                          DNMB3D:
007700  104000                          EMT                     ;COND. CODES INCORRECT
007702  005004                  DNMB3E: CLR     R4              ;R4=0
007704  022714  125125                  CMP     #125125,(R4)    ;CHECK DEST. DATA
007710  001401                          BEQ     TS142
007712  104000                          EMT                     ;DEST. DATA MODIFIED

                                ;************************************************************************
                                ;TEST 142       TEST DEST. MODE 4 W/DOP NON-MODIFYING INST.
                                ;************************************************************************
007714                          TS142:
007714  005000                          CLR     R0              ;R0=0
007716  005010                          CLR     (R0)            ;LOC. 0=0
007720  052710  125252                  BIS     #125252,(R0)    ;LOC. 0=125125
007724  052700  000002                  BIS     #2,R0           ;R0=2
007730  000277                          SCC                     ;SET ALL COND. CODE BITS
007732  032740  020000                  BIT     #20000,-(R0)    ;TRY DOPNM W/ MODE 4
007736  100403                          BMI     DNM4A           ;BR TO ERROR IF N-BIT SET
007740  102402                          BVS     DNM4A           ;BR TO ERROR IF V-BIT SET
007742  103001                          BCC     DNM4A           ;BR TO ERROR IF C-BIT CHAR
007744  001001                          BNE     DNM4B
007746                          DNM4A:
007746  104000                          EMT                     ;COND. CODES INCORRECT
007750  005700                  DNM4B:  TST     R0              ;CHECK DEST. REGISTER
007752  001401                          BEQ     DNM4C
007754  104000                          EMT                     ;DEST. REGISTER NOT DECREMENTED BY 2
007756  022737  125252  000000  DNM4C:  CMP     #125252,@#0     ;CHECK DEST. DATA
007764  001401                          BEQ     TS143
007766  104000                          EMT                     ;DEST. DATA MODIFIED

                                ;************************************************************************
                                ;TEST 143       TEST DEST. MODE 4-BYTE W/ DOP NON-MODIFYING INST.
                                ;************************************************************************
007770                          TS143:
007770  005000                          CLR     R0              ;R0=0
007772  005010                          CLR     (R0)            ;LOC. 0=0
007774  052710  052652                  BIS     #52652,(R0)     ;LOC. 0=52652
010000  052700  000002                  BIS     #2,R0           ;R0=2
010004  000257                          CCC                     ;COND. CODES=0
010006  132740  000201                  BITB    #201,-(R0)      ;TRY DOPNM INST W/MODE 4 ODD BYTE
010012  102403                          BVS     DNMB4A          ;BR TO ERROR IF V BIT SET
010014  001402                          BEQ     DNMB4A  ;BR TO ERROR IF Z BIT SET
010016  103401                          BCS     DNMB4A          ;BR TO ERROR IF C BIT SET
010020  001001                          BNE     DNMB4B
010022                          DNMB4A:
010022  104000                          EMT                     ;COND. CODES INCORRECT
010024  022700  000001          DNMB4B: CMP     #1,R0           ;CHECK DEST. REGISTER
010030  001401                          BEQ     DNMB4C
010032  104000                          EMT                     ;DEST REG. NOT DECREMENTED BY 1
010034  132740  000201          DNMB4C: BITB    #201,-(R0)      ;TRY DOPNM INST. W/MODE 4 EVEN BYTE
010040  001401                          BEQ     DNMB4D          ;BR TO ERROR IF Z-BIT SET
010042  100401                          BMI     DNMB4E
010044                          DNMB4D:
```

```
010044  104000                          EMT                             ;COND. CODES INCORRECT
010046  005700                  DNMB4E: TST     R0                      ;CHECK DEST. REGISTER
010050  001401                          BEQ     DNMB4F
010052  104000                          EMT                             ;DEST. REG. NOT DECREMENTED BY 1
010054  022710  052652          DNMB4F: CMP     #52652,(R0)             ;CHECK DESTINATION DATA
010060  001401                          BEQ     TS144
010062  104000                          EMT                             ;DEST. DATA MODIFIED

                                ;*******************************************************************************
                                ;TEST 144       TEST DEST MODE 5 W/DOP NON-MODIFYING INST.
                                ;*******************************************************************************
010064                          TS144:
010064  005000                          CLR     R0                      ;R0=0
010066  005010                          CLR     (R0)                    ;LOC 0=0
010070  052710  100000                  BIS     #100000,(R0)            ;LOC. 0=100000
010074  052700  000402                  BIS     #402,R0                 ;R0=2
010100  000277                          SCC                             ;SET ALL COND. CODE BITS
010102  032750  100000                  BIT     #100000,@-(R0)          ;TRY DOPNM W/MODE 5
010106  102403                          BVS     DNM5A                   ;BR TO ERROR IF V-BIT SET
010110  103002                          BCC     DNM5A                   ;BR TO ERROR IF C-BIT CLEAR
010112  001401                          BEQ     DNM5A                   ;BR TO ERROR IF Z-BIT SET
010114  100401                          BMI     DNM5B
010116                          DNM5A:
010116  104000                          EMT                             ;COND. CODES INCORRECT
010120  022700  000400          DNM5B:  CMP #400,R0                     ;CHECK DEST. REGISTER
010124  001401                          BEQ     DNM5C
010126  104000                          EMT                             ;DEST. REGISTER NOT DECREMENTED BY 2
010130  022737  100000  000000  DNM5C:  CMP     #100000,@#0             ;CHECK DESTINATION DATA
010136  001401                          BEQ     TS145
010140  104000                          EMT                             ;DEST. DATA INCORRECTLY MODIFIED

                                ;*******************************************************************************
                                ;TEST 145       TEST DEST. MODE 6 W/DOP NON-MODIFYING INST.
                                ;*******************************************************************************
010142                          TS145:
010142  005000                          CLR     R0                      ;R0=0
010144  005010                          CLR     (R0)                    ;LOC> 0=0
010146  052710  000001                  BIS     #1,(R0)                 ;LOC. 0=1
010152  005100                          COM     R0                      ;R0=-1 C-BIT=1
010154  032760  000001  000001          BIT     #1,1(R0)                ;TRY DOPNM W/MODE 6
010162  001403                          BEQ     DNM6A                   ;BR TO ERROR IF Z-BIT SET
010164  102402                          BVS     DNM6A                   ;BR TO ERROR IF V-BIT SET
010166  103001                          BCC     DNM6A                   ;BR TO ERROR IF C-BIT CLEAR
010170  100001                          BPL     DNM6B
010172                          DNM6A:
010172  104000                          EMT                             ;COND CODES INCORRECT
010174  022700  177777          DNM6B:  CMP     #-1,R0                  ;CHECK DEST. REGISTER
010200  001401                          BEQ     DNM6C
010202  104000                          EMT                             ;DEST. REGISTER MODIFIED
010204  022737  000001  000000  DNM6C:  CMP     #1,@#0                  ;CHECK DEST. DATA
010212  001401                          BEQ     TS146
010214  104000                          EMT                             ;DEST. DATA MODIFIED

                                ;*******************************************************************************
                                ;TEST 146       TEST DEST MODE 7 W/DOP NON-MODIFYING INST.
                                ;*******************************************************************************
```

```
010216                                TS146:
010216  005000                                CLR     R0              ;R0=0
010220  005010                                CLR     (R0)            ;LOC. 0=0 C-BIT=0
010222  052710  125125                        BIS     #125125,(R0)    ;LOC. 0=125125
010226  052700  000001                        BIS     #1,R0           ;R0=1
010232  132770  000125  000403                BITB    #125,@403(R0)   ;TRY DOPNM W/MODE 7
010240  102403                                BVS     DNM7A           ;BR TO ERROR IF V-BIT SET
010242  100402                                BMI     DNM7A           ;BR TO ERROR IF N-BIT SET
010244  103401                                BCS     DNM7A           ;BR TO ERROR IF C-BIT SET
010246  001401                                BEQ     DNM7B
010250                                DNM7A:
010250  104000                                EMT                     ;COND. CODES INCORRECT
010252  022700  000001                DNM7B:  CMP     #1,R0           ;CHECK DEST. REGISTER
010256  001401                                BEQ     DNM7C
010260  104000                                EMT                     ;DESTINATION REGISTER MODIFIED
010262  022737  125125  000000        DNM7C:  CMP     #125125,@#0     ;CHECK DEST. DATA
010270  001401                                BEQ     TS147
010272  104000                                EMT                     ;DEST. DATA INCORRECT

                                      ;*************************************************************************
                                      ;
                                      ;       THIS TEST VERIFIES THE MOV DESTINATION MODE 1 INSTRUCTION.
                                      ;DATA IS SET IN R0 USING SOP INSTRUCTIONS AND THEN MOVED TO LOC. 0
                                      ;USING MOV SRC MODE 0, DEST. MODE 1.
                                      ;
                                      ;*************************************************************************
                                      ;TEST 147       TEST MOV DESTINATION MODE 1
                                      ;*************************************************************************
010274                                TS147:
010274  005000                                CLR     R0              ;R0=0
010276  005010                                CLR     (R0)            ;LOC. 0=0
010300  005100                                COM     R0              ;R0=-1
010302  005004                                CLR     R4              ;R4 POINTS TO LOC. 0
010304  010014                                MOV     R0,(R4)         ;TRY MOVE MODE 0,1
010306  102402                                BVS     MDM1A           ;BR TO ERROR IF V SET
010310  001401                                BEQ     MDM1A           ;BR TO ERROR IF Z SET
010312  100401                                BMI     MDM1B
010314                                MDM1A:
010314  104000                                EMT                     ;CONDITION CODE NOT CORRECT
010316  005704                        MDM1B:  TST     R4
010320  001401                                BEQ     TS150
010322  104000                                EMT                     ;DESTINATION REGISTER INCORRECTLY ALTERED

                                      ;*************************************************************************
                                      ;
                                      ;       THIS TEST VERIFIES THE MOV DESTINATION MODE 2 INSTRUCTION.
                                      ;       DATA IS SET IN R0 USING SOP INSTRUCTIONS AND THEN MOVED
                                      ;       TO LOCATION 0 USING MOV SRC MODE 0, DEST. MODE 1.
                                      ;
                                      ;*************************************************************************
                                      ;TEST 150       TEST MOV DESTINATION MODE 2
                                      ;*************************************************************************
010324                                TS150:
010324  005000                                CLR     R0              ;R0=0
010326  005001                                CLR     R1              ;R1=0
010330  005010                                CLR     (R0)            ;LOC.0=0
```

```
010332  005110                             COM     (R0)            ;LOC. 0= 1
010334  010120                             MOV     R1,(R0)+        ;TRY MOVE MODE 0,2
010336  100402                             BMI     MDM2A           ;BR TO ERROR IF N SET
010340  102401                             BVS     MDM2A           ;BR TO ERROR IF V SET
010342  001401                             BEQ     MDM2B
010344                             MDM2A:
010344  104000                             EMT                     ;CC'S INCORRECT
010346  005300                     MDM2B:  DEC     R0
010350  005300                             DEC     R0
010352  001401                             BEQ     MDM2D
010354                             MDM2C:
010354  104000                             EMT                     ;DESTINATION REGISTER NOT INCREMENTED PROPERLY
010356  005737  000000             MDM2D:  TST     @#0
010362  001401                             BEQ     TS151
010364  104000                             EMT                     ;DESTINATION DATA INCORRECT

                                   ;*****************************************************************************
                                   ;
                                   ;       THIS TEST VERIFIES DESTINATION MODE 2 W/MOVB INSTS. TWO DIFFERENT MOVB
                                   ;INSTRUCTIONS ARE USED TO MOVE A TEST PATTERN FIRST TO BYTE 0 THEN TO BYTE 1.
                                   ;
                                   ;*****************************************************************************
                                   ;TEST 151       TEST MOV-BYTE DESTINATION MODE 2
                                   ;*****************************************************************************
010366                             TS151:
010366  005000                             CLR     R0              ;R0=0
010370  005010                             CLR     (R0)            ;LOC. 0=0
010372  112720  000125                     MOVB    #125,(R0)+      ;TRY DESTINATION MODE 2 W/EVEN BYTE
010376  102402                             BVS     MBDM2A          ;BR TO ERROR IF V SET
010400  001401                             BEQ     MBDM2A          ;BR TO ERROR IF Z SET
010402  100001                             BPL     MBDM2B
010404                             MBDM2A:
010404  104000                             EMT                     ;CC'S INCORRECT
010406  022700  000001             MBDM2B: CMP     #1,R0
010412  001401                             BEQ     MBDM2C
010414  104000                             EMT                     ;REGISTER NOT INCREMENTED BY ONE
010416  112720  000252             MBDM2C: MOVB    #252,(R0)+      ;TRY DESTINATION MODE 2 W/ODD BYTE
010422  102402                             BVS     MBDM2D
010424  001401                             BEQ     MBDM2D
010426  100401                             BMI     MBDM2E
010430                             MBDM2D:
010430  104000                             EMT                     ;CC'S NOT SET CORRECT
010432  022700  000002             MBDM2E: CMP     #2,R0
010436  001401                             BEQ     MBDM2F
010440  104000                             EMT                     ;REGISTER NOT INCREMENTED BY ONE
010442  022737  125125  000000     MBDM2F: CMP     #125125,@#0     ;CHECK DATA
010450  001401                             BEQ     TS152
010452  104000                             EMT                     ;DESTINATION DATA INCORRECT

                                   ;*****************************************************************************
                                   ;
                                   ;       THIS TEST VERIFIES MOV DESTINATION MODE 3. R0 IS USED TO PICK UP
                                   ;AN ADDRESS AT LOC. 400. LOC 400 POINTS TO LOC. 0 THE EFFECTIVE DEST. ADDR.. ALSO, MOVB
                                   ;INST. ARE USED W/ EVEN AND ODD BYTES TO CHECK MOV BYTES INST AND MODE 37 DESTINATIONS.
                                   ;*****************************************************************************
                                   ;TEST 152       TEST MOV(B) DESTINATION MODE 3
```

```
                                              ;*************************************************************************
010454                                        TS152:
010454  012700  000400                                MOV     #400,R0          ;R0=400
010460  005010                                        CLR     (R0)             ;LOC. 400 POINTS TO LOC. 0
010462  005037  000000                                CLR     @#0              ;LOC. 0=0
010466  012730  125252                                MOV     #125252,@(R0)+   ;TRY MOV DESTINATION MODE 2
010472  102402                                        BVS     MDM3A            ;BR TO ERROR IF V SET
010474  001401                                        BEQ     MDM3A            ;BR TO ERROR IF Z SET
010476  100401                                        BMI     MDM3B
010500                                        MDM3A:
010500  104000                                        EMT                      ;CC'S INCORRECT
010502  022700  000402                        MDM3B:  CMP     #402,R0          ;CHECK DEST. MODE REGISTER
010506  001401                                        BEQ     MDM3C
010510  104000                                        EMT                      ;REGISTER NOT INCREMENTED BY 2
010512  022737  125252  000000                MDM3C:  CMP     #125252,@#0      ;CHECK DESTINATION DATA
010520  001401                                        BEQ     MDM3D
010522  104000                                        EMT                      ;DESTINATION DATA INCORRECT
010524  112737  000125  000000                MDM3D:  MOVB    #125,@#0         ;TRY MOVB DESTINATION MODE 2 EVEN BYTE
010532  022737  125125  000000                        CMP     #125125,@#0      ;CHECK DATA
010540  001401                                        BEQ     MDM3E
010542  104000                                        EMT                      ;DESTINATION DATA INCORRECT
010544  112737  000525  000001                MDM3E:  MOVB    #525,@#1         ;TRY MOVB DESTINATION MODE 2 ODD BYTE
010552  022737  052525  000000                        CMP     #52525,@#0       ;CHECK DATA
010560  001401                                        BEQ     TS153
010562  104000                                        EMT                      ;

                                              ;*************************************************************************
                                              ;
                                              ;       THIS TEST VERIFIES THE MOV DESTINATION MODE 4 INSTRUCTION.
                                              ;SOP INSTRUCTIONS ON R0 ARE USED TO CLEAR TARGET LOCATION 0.
                                              ;R4 IS USED AS THE MODE 4 ADDRESSING REGISTER, AND
                                              ;CONDITIONAL BRANCHES ARE USED TO VERIFY THE DATA.
                                              ;
                                              ;*************************************************************************
                                              ;TEST 153        TEST MOV DESTINATION MODE 4
                                              ;*************************************************************************
010564                                        TS153:
010564  005000                                        CLR     R0               ;R0=0
010566  005010                                        CLR     (R0)             ;LOC 0=0
010570  012704  000002                                MOV     #2,R4            ;R4=2
010574  012744  012345                                MOV     #12345,-(R4)     ;TRY MOV DEST. MODE 4
010600  102402                                        BVS     MDM4A            ;BR TO ERROR IF V-BIT SET
010602  001401                                        BEQ     MDM4A            ;BR TO ERROR IF Z-BIT SET
010604  100001                                        BPL     MDM4B
010606                                        MDM4A:
010606  104000                                        EMT                      ;CC'S NOT CORRECT
010610  005704                                MDM4B:  TST     R4               ;CHECK DECREMENTING OF MODE 4 REG.
010612  001401                                        BEQ     MDM4C
010614  104000                                        EMT                      ;DESTINATION MODE REGISTER NOT DECREMENTED BY 2
010616  022710  012345                        MDM4C:  CMP     #12345,(R0)      ;CHECK DESTINATION DATA
010622  001401                                        BEQ     TS154
010624  104000                                        EMT                      ;DESTINATION DATA INCORRECT

                                              ;*************************************************************************
                                              ;
                                              ;       THIS TEST VERIFIES THE MOVB DESTINATION MODE 4 INSTRUCTION
```

```
                                   ;        ON BOTH ODD AND EVEN BYTES. SOP INSTRUCTIONS ON R4 ARE
                                   ;        USED TO CLEAR TARGET LOCATION 0. R0 IS USED AS THE MODE 4
                                   ;        ADDRESSING REGISTER, AND CMP AND CONDITIONAL BRANCH
                                   ;        INSTRUCTIONS ARE USED TO VERIFY THE DATA.
                                   ;
                                   ;**********************************************************************
                                   ;TEST 154        TEST MOVB DESTINATION MODE 4
                                   ;**********************************************************************
 010626                            TS154:
 010626  005004                            CLR     R4              ;R4=0
 010630  005014                            CLR     (R4)            ;LOC. 0=0
 010632  012700  000002                    MOV     #2,R0           ;R0 = 2
 010636  112740  125125                    MOVB    #125125,-(R0)   ;TRY MOVB DEST. MODE 4-ODD BYTE
 010642  020027  000001                    CMP     R0,#1           ;CHECK THAT DEST. REG. WAS DECREMENTED
 010646  001401                            BEQ     MBDM4A
 010650  104000                            EMT                     ;DESTINATION REG. NOT DECREMENTED BY 1
 010652  021427  052400            MBDM4A: CMP     (R4),#52400     ;CHECK DEST. DATA
 010656  001401                            BEQ     MBDM4B
 010660  104000                            EMT                     ;DEST. DATA NOT CORRECT
 010662  112740  125125            MBDM4B: MOVB    #125125,-(R0)   ;TRY MOVB DEST. MODE 4--EVEN BYTE
 010666  102402                            BVS     MBDM4C          ;BR. TO ERROR IF V-BIT SET
 010670  001401                            BEQ     MBDM4C          ;BR TO ERROR IF Z-BIT SET
 010672  100001                            BPL     MBDM4D
 010674                            MBDM4C:
 010674  104000                            EMT                     ;COND. CODES INCORRECT
 010676  005700                    MBDM4D: TST     R0              ;CHECK MODE 4 DEST. REGISTER
 010700  001401                            BEQ     MBDM4E
 010702  104000                            EMT                     ;DESTINATION REG NOT DECREMENTED BY 1
 010704  021427  052525            MBDM4E: CMP     (R4),#52525     ;CHECK DEST. DATA
 010710  001401                            BEQ     TS155
 010712  104000                            EMT                     ;DESTINATION DATA INCORRECT

                                   ;**********************************************************************
                                   ;
                                   ;        THIS TEST VERIFIES THE MOV DESTINATION MODE 5 AND THE MOVB
                                   ;DESTINATION MODE 5 - EVEN BYTE INSTRUCTIONS. R4 IS A
                                   ;POINTER TO TARGET LOCATION 0 AND R0 IS SETUP TO
                                   ;POINT TO LOCATION 376 FOR THE MOV, AND LOCATION 404 FOR
                                   ;THE MOVB INSTRUCTIONS. CMP INSTRUCTIONS ARE USED TO VERIFY
                                   ;PROPER ADDRESSING AND DATA.
                                   ;
                                   ;**********************************************************************
                                   ;TEST 155        TEST MOV DESTINATION MODE 5
                                   ;**********************************************************************
 010714                            TS155:
 010714  005004                            CLR     R4              ;R4=0
 010716  005014                            CLR     (R4)            ;LOC. 0 = 0
 010720  012700  000400                    MOV     #400,R0         ;R0=400
 010724  012750  004321                    MOV     #4321,@-(R0)    ;TRY MOV DEST. MODE 5
 010730  102402                            BVS     MDM5A           ;BR TO ERROR IF V-BIT SET
 010732  001401                            BEQ     MDM5A           ;BR TO ERROR IF Z-BIT SET
 010734  100001                            BPL     MDM5B
 010736                            MDM5A:
 010736  104000                            EMT                     ;COND. CODES INCORRECT
 010740  022700  000376            MDM5B:  CMP     #376,R0         ;CHECK MODE 5 REG. WAS DECREMENTED
 010744  001401                            BEQ     MDM5C
```

```
010746  104000                          EMT                             ;MODE 5 REGISTER NOT DECREMENTED BY 2
010750  022714  004321          MDM5C:  CMP     #4321,(R4)              ;CHECK DEST. DATA
010754  001401                          BEQ     MDM5D
010756  104000                          EMT                             ;DEST. DATA INCORRECT
010760  012700  000406          MDM5D:  MOV     #406,R0                 ;R0=406
010764  112750  000377                  MOVB    #377,@-(R0)             ;TRY MOV DEST. MODE 5 --EVEN BYTE
010770  022700  000404                  CMP     #404,R0                 ;CHECK MODE 5 REG.
010774  001401                          BEQ     MDM5E
010776  104000                          EMT                             ;MODE 5 REGISTER NOT DECREMENTED BY 2
011000  022714  177721          MDM5E:  CMP     #177721,(R4)            ;CHECK DEST. DATA
011004  001401                          BEQ     TS156
011006  104000                          EMT                             ;DEST. DATA INCORRECT

                                ;*****************************************************************************
                                ;
                                ;       THIS TEST VERIFIES THE MOV DESTINATION MODE 6 AND MOVB - EVEN BYTE
                                ;DESTINATION MODE 6 INDTRUCTIONS. R0 IS USED TO SETUP TARGET LOC.0
                                ;FOR BOTH TESTS. PATTERNS OF ONES AND ZEROES ARE MOVED INTO LOC.0
                                ;BY MODE 6 INSTRUCTIONS, AND CMP INSTRUCTIONS ARE USED TO VERIFY
                                ;PROPER ADDRESSING AND DATA.
                                ;
                                ;*****************************************************************************
                                ;TEST 156       TEST MOV DESTINATION MODE 6
                                ;*****************************************************************************
011010                          TS156:
011010  005000                          CLR     R0                      ;R0=0
011012  005010                          CLR     (R0)                    ;LOC. 0=0
011014  005200                          INC     R0                      ;R0=1
011016  012760  052525  177777          MOV     #052525,-1(R0)          ;TRY MOV DEST. MODE 6
011024  102402                          BVS     MDM6A                   ;BR TO ERROR IF V-BIT SET
011026  001401                          BEQ     MDM6A                   ;BR TO ERROR IF Z-BIT SET
011030  100001                          BPL     MDM6B
011032                          MDM6A:
011032  104000                          EMT                             ;COND. CODES INCORRECT
011034  022700  000001          MDM6B:  CMP     #1,R0                   ;CHECK DEST. REGISTER UNALTERED
011040  001401                          BEQ     MDM6C
011042  104000                          EMT                             ;DEST. REGISTER INCORRECTLY ALTERED
011044  022737  052525  000000  MDM6C:  CMP     #52525,@#0              ;CHECK DEST. DATA
011052  001401                          BEQ     MDM6D
011054  104000                          EMT                             ;DEST. DATA INCORRECT
011056  012700  000002          MDM6D:  MOV     #2,R0                   ;R0=2
011062  112760  000377  177777          MOVB    #377,-1(R0)             ;TRY MOVB DEST. MODE 6
011070  022700  000002                  CMP     #2,R0                   ;CHECK DEST. REGISTER UNALTERED
011074  001401                          BEQ     MDM6E
011076  104000                          EMT                             ;DEST. REGISTER INCORRECTLY ALTERED
011100  022737  177525  000000  MDM6E:  CMP     #177525,@#0             ;CHECK DEST. DATA
011106  001401                          BEQ     TS157
011110  104000                          EMT                             ;DEST. DATA INCORRECT

                                ;*****************************************************************************
                                ;
                                ;       THIS TEST VERIFIES THE MOV DESTINATION MODE 7 AND MOVB - ODD BYTE
                                ;DESTINATION MODE 7 INSTRUCTIONS. R4 POINTS TO TARGET LOC.0 AND R0
                                ;IS USED AS THE MODE 7 ADDRESSING REGISTER. CMP INSTRUCTIONS ARE
                                ;USED TO VERIFY PROPER ADDRESSING AND DATA.
                                ;
```

```
                                       ;*******************************************************************************
                                       ;TEST 157        TEST MOV DESTINATION MODE 7
                                       ;*******************************************************************************
011112                                 TS157:
011112  005004                                 CLR     R4              ;R4=0
011114  005014                                 CLR     (R4)            ;LOC.0=0
011116  012700  000403                         MOV     #403,R0         ;R0=403
011122  012770  070707  177777                 MOV     #70707,@-1(R0)  ;TRY MOV W/DEST MODE 7
011130  102402                                 BVS     MDM7A           ;BR. TO ERROR IF V-BIT SET
011132  001401                                 BEQ     MDM7A           ;BR TO ERROR IF Z-BIT SET
011134  100001                                 BPL     MDM7B
011136                                 MDM7A:
011136  104000                                 EMT                     ;COND. CODES INCORRECT
011140  022700  000403                 MDM7B:  CMP     #403,R0         ;CHECK DEST. REGISTER
011144  001401                                 BEQ     MDM7C
011146  104000                                 EMT                     ;DEST. REGISTER INCORRECTLY ALTERED
011150  022737  070707  000000         MDM7C:  CMP     #70707,@#0      ;CHECK DEST. DATA
011156  001401                                 BEQ     MDM7D
011160  104000                                 EMT                     ;DEST. DATA INCORRECT
011162  112770  107070  000001         MDM7D:  MOVB    #107070,@1(R0)  ;TRY MOVB W/DEST MODE 7--ODD BYTE
011170  022700  000403                         CMP     #403,R0         ;CHECK MODE 7 DEST. REG.
011174  001401                                 BEQ     MDM7E
011176  104000                                 EMT                     ;DEST. DATA INCORRECT
011200  022737  034307  000000         MDM7E:  CMP     #34307,@#0      ;CHECK DEST. DATA
011206  001401                                 BEQ     TS160
011210  104000                                 EMT                     ;DESTINATION DATA INCORRECT

                                       ;*******************************************************************************
                                       ;
                                       ;       THIS TEST VERIFIES MODE 4 DOUBLE OPERAND INSTRUCTIONS.
                                       ;THE TEST USES MODE 4 ADDRESSING WITH REGISTER 0 TO MOVE THRU A
                                       ;TABLE OF OPERANDS.  THE TABLE OF OPERANDS AND THE WORK LOCATION IS
                                       ;STORED FOLLOWING THE TEST CODE.  A SERIES OF 5 DOP INSTRUCTIONS UTILIZES
                                       ;THE DATA IN THE TABLE TO CYCLE THE WORK LOCATION THRU A SET OF
                                       ;VALUE.  THE DATA HAS BEEN CHOSEN TO INSURE THAT NO SINGLE ERROR WILL
                                       ;GO UNDETECTED.  WORD AND BYTE INSTRUCTION ACCESSING BOTH EVEN AND
                                       ;ODD ADDRESSES ARE USED IN THE TEST.  THE LISTING SHOWS THE
                                       ;EXPECTED INTERMEDIATE RESULT AS EACH INSTRUCTION IS EXECUTED.
                                       ;
                                       ;*******************************************************************************
                                       ;TEST 160        TEST MODE 4 W/ DOP INSTS.
                                       ;*******************************************************************************
011212                                 TS160:
011212  012700  011256                         MOV     #TBL1,R0        ;INITIALIZE R0
011216  014037  011256                         MOV     -(R0),@#TBL1    ;TBL1=125252
011222  064037  011256                         ADD     -(R0),@#TBL1    ;TBL1=000377
011226  144037  011256                         BICB    -(R0),@#TBL1    ;TBL1=000252
011232  154037  011257                         BISB    -(R0),@#TBL1+1  ;TBL1=125252
011236  024037  011256                         CMP     -(R0),@#TBL1    ;CHECK RESULT
011242  001406                                 BEQ     TS161
011244                                 DOP4:
011244  104000                                 EMT                     ;RESULT OF MODE 4 INSTS. INCORRECT

011246  125252                                 125252
011250  052652                                 52652
011252  053125                                 53125
```

```
011254  125252                          125252
011256  000000                  TBL1:   0

                                ;*******************************************************************************
                                ;
                                ;       THIS TEST VERIFIES MODE 5 DOUBLE OPERAND INSTRUCTIONS.
                                ;THE TEST USES AN ADDRESS TABLE STORED FOLLOWING THE TEST CODE.
                                ;THIS TABLE IS SIMPLY A TABLE OF ADDRESS POINTERS WHICH ADDRESS
                                ;THE DATA TABLE USED IN THE PREVIOUS TEST.  THE TEST IS IDENTICAL TO
                                ;THE PREVIOUS TEST EXCEPT THE DATA IS REFERENCED USING THIS ADDRESS
                                ;TABLE AND MODE 5 ADDRESSING.  (SEE PREVIOUS TEST).
                                ;
                                ;*******************************************************************************
                                ;TEST 161       TEST MODE 5 W/ DOP INSTS.
                                ;*******************************************************************************
011260                          TS161:
011260  012700  011326                  MOV     #TBL2+2,R0      ;INITIALIZE R0
011264  015037  011256                  MOV     @-(R0),@#TBL1   ;TBL1=125252
011270  065037  011256                  ADD     @-(R0),@#TBL1   ;TBL1=000377
011274  145037  011256                  BICB    @-(R0),@#TBL1   ;TBL1=000252
011300  155037  011257                  BISB    @-(R0),@#TBL1+1 ;TBL1=125252
011304  025037  011256                  CMP     @-(R0),@#TBL1   ;CHECK RESULT
011310  001406                          BEQ     TS162
011312                          DOP5:
011312  104000                          EMT                     ;RESULT OF MODE 5 INSTS. INCORRECT
011314  011246                          TBL1-10
011316  011250                          TBL1-6
011320  011251                          TBL1-5
011322  011252                          TBL1-4
011324  011254                  TBL2:   TBL1-2
                                ;*******************************************************************************
                                ;
                                ;       THIS TEST VERIFIES MODE 6 DOUBLE OPERAND INSTRUCTIONS.
                                ;IT USES THE SAME DATA AS THAT USED IN THE MODE 4 TESTS.
                                ;THIS TIME THE DATA IS ACCESSED USING MODE 6.  R0 IS SET
                                ;TO POINT TO THE MIDDLE OF THE TABLE.  THE TABLE IS ACCESSED FROM
                                ;BOTTOM TO TOP BY VARYING THE OFFSET IN THE MODE 6 INSTRUCTIONS.
                                ;THE DATA RESULTS ARE IDENTICAL TO THOSE EXPECTED IN THE MODE 4
                                ;TESTS.
                                ;
                                ;*******************************************************************************
                                ;TEST 162       TEST MODE 6 W/ DOP INSTS.
                                ;*******************************************************************************
011326                          TS162:
011326  012700  011252                  MOV     #TBL1-4,R0      ;INITIALIZE R0
011332  016037  000002  011256          MOV     2(R0),@#TBL1    ;TBL1=125252
011340  066037  000000  011256          ADD     0(R0),@#TBL1    ;TBL1=000377
011346  146037  177777  011256          BICB    -1(R0),@#TBL1   ;TBL1=000252
011354  156037  177776  011257          BISB    -2(R0),@#TBL1+1 ;TBL1=125252
011362  026037  177774  011256          CMP     -4(R0),@#TBL1   ;CHECK RESULT
011370  001401                          BEQ     TS163
011372  104000                          EMT                     ;RESULT OF MODE 6 INSTS. INCORRECT
                                ;*******************************************************************************
                                ;
                                ;       THIS TEST VERIFIES MODE 7 DOUBLE OPERAND INSTRUCTIONS.
                                ;THIS TEST USES THE SAME ADDRESS TABLE AND DATA TABLE USED BY
```

```
                                        ;THE MODE 5 TESTS.  THIS TIME THE DATA IS ACCESSED USING MODE 7.
                                        ;R0 IS SET TO POINT TO THE MIDDLE OF THE ADDRESS TABLE IN THE MODE 5
                                        ;TEST.  THE TABLE IS ACCESSED FROM BOTTOM TO TOP BY VARYING THE OFFSET
                                        ;IN THE MODE 7 INSTRUCTIONS.  THE DATA RESULTS ARE IDENTICAL TO
                                        ;THOSE EXPECTED IN THE MODE 5 TESTS.
                                        ;
                                        ;******************************************************************************
                                        ;TEST 163       TEST MODE 7 W/ DOP INSTS.
                                        ;******************************************************************************
 011374                                 TS163:
 011374  012700  011320                         MOV     #TBL2-4,R0      ;INITIALIZE R0
 011400  017037  000004  011256                 MOV     @4(R0),@#TBL1   ;TBL1=125252
 011406  067037  000002  011256                 ADD     @2(R0),@#TBL1   ;TBL1=000377
 011414  147037  000000  011256                 BICB    @0(R0),@#TBL1   ;TBL1=000252
 011422  157037  177776  011257                 BISB    @-2(R0),@#TBL1+1                ;TBL1=125252
 011430  027037  177774  011256                 CMP     @-4(R0),@#TBL1          ;CHECK RESULT
 011436  001401                                 BEQ     TS164
 011440  104000                                 EMT                     ;RESULT OF MODE 7 INSTS INCORRECT
                                        ;******************************************************************************
                                        ;
                                        ;       THIS TEST VERIFIES THE ROTATE MODE 0 INSTRUCTIONS.
                                        ;R0 IS LOADED WITH A DATA PATTERN, THE C-BIT IS LOADED, AND
                                        ;AN ROL INSTRUCTION IS EXECUTED WITH MODE 0.  THE OPERATION IS CHECKED
                                        ;BY TESTING THE RESULTING DATA AND THE STATE OF THE C AND V BITS.
                                        ;NEXT, THE SAME PROCEDURE IS EXECUTED TO TEST MODE 0 BYTE INSTRUCTIONS.
                                        ;
                                        ;******************************************************************************
                                        ;TEST 164       TEST ROTATE INSTRUCTIONS OF MODE 0
                                        ;******************************************************************************
 011442                                 TS164:
 011442  012700  125252                         MOV     #125252,R0      ;INITIALIZE DATA
 011446  000261                                 SEC                     ;SET C-BIT
 011450  006100                                 ROL     R0              ;TRY ROL W/ MODE 0
 011452  102004                                 BVC     ROT0A           ;CC=0011
 011454  103003                                 BCC     ROT0A
 011456  022700  052525                         CMP     #052525,R0      ;CHECK DATA
 011462  001401                                 BEQ     ROT0B
 011464                                 ROT0A:
 011464  104000                                 EMT                     ;ROL MODE 0 FAILED
 011466  012700  125252                 ROT0B:  MOV     #125252,R0      ;INITIALIZE DATA
 011472  000261                                 SEC                     ;SET C-BIT
 011474  106100                                 ROLB    R0              ;TRY ROL W/ MODE 0 EVEN BYTE
 011476  102004                                 BVC     ROT0C           ;CC=0011
 011500  103003                                 BCC     ROT0C
 011502  022700  125125                         CMP     #125125,R0      ;CHECK DATA
 011506  001401                                 BEQ     TS165
 011510                                 ROT0C:
 011510  104000                                 EMT                     ;ROLB MODE 0 FAILED

                                        ;******************************************************************************
                                        ;
                                        ;       THIS TEST VERIFIES THE ROTATE MODE 1 INSTRUCTIONS.
                                        ;THE DATA TO BE ROTATED IS IN LOC 0.  R0 IS USED AS THE
                                        ;ADDRESSING REGISTER.  THE C-BIT IS LOADED AND AN ROL IS EXECUTED.
                                        ;THE RESULTS ARE CHECKED BY COMPARING THE DATA RESULTS AND TESTING
                                        ;THE C AND V BITS.  THIS PROCEDURE IS THEN REPEATED TWICE MORE
```

```
                                        ;TO TEST THE BYTE ROTATES.  FIRST ON BYTE 0, THEN ON BYTE 1.
                                        ;
                                        ;*******************************************************************************
                                        ;TEST 165       TEST ROTATE INSTRUCTIONS W/ MODE 1
                                        ;*******************************************************************************
011512                                  TS165:
011512  005000                                  CLR     R0                      ;POINT TO LOC. 0
011514  012710  052525                          MOV     #52525,(R0)             ;INITIALIZE DATA
011520  000241                                  CLC                             ;CLEAR C-BIT
011522  006110                                  ROL     (R0)                    ;TRY ROL W/ MODE 1
011524  102005                                  BVC     ROT1A                   ;CC=1010
011526  103404                                  BCS     ROT1A
011530  023727  000000  125252                  CMP     @#0,#125252             ;CHECK RESULT
011536  001401                                  BEQ     ROT1B
011540                                  ROT1A:
011540  104000                                  EMT                             ;ROL MODE 1 FAILED
011542  000261                          ROT1B:  SEC
011544  012710  125252                          MOV     #125252,(R0)            ;INITIALIZE DATA
011550  106110                                  ROLB    (R0)                    ;TRY ROLB W/ MODE 1 EVEN BYTE
011552  102005                                  BVC     ROT1C                   ;CC=1011
011554  103004                                  BCC     ROT1C
011556  022737  125125  000000                  CMP     #125125,@#0             ;TEST RESULT
011564  001401                                  BEQ     ROT1D
011566                                  ROT1C:
011566  104000                                  EMT                             ;ROLB W/ MODE 1 EVEN BYTE FAILED
011570  012710  125252                  ROT1D:  MOV     #125252,(R0)
011574  005000                                  CLR     R0                      ;POINT TO ODD BYTE
011576  005200                                  INC     R0
011600  000261                                  SEC                             ;SET C-BIT
011602  106110                                  ROLB    (R0)                    ;TRY ROLB W/ MODE 1 ODD BYTE
011604  102005                                  BVC     ROT1E                   ;CC=0011
011606  103004                                  BCC     ROT1E
011610  022737  052652  000000                  CMP     #052652,@#0             ;CHECK DATA
011616  001401                                  BEQ     TS166
011620                                  ROT1E:
011620  104000                                  EMT                             ;ROLB W/ MODE 1 ODD BYTE FAILED

                                        ;*******************************************************************************
                                        ;
                                        ;       THIS TEST VERIFIES MODE 2 ROTATE INSTRUCTIONS.
                                        ;THE SAME PROCEDURE AS IN THE OTHER ROTATE TESTS ARE USED.  R0
                                        ;IS USED AS THE ADDRESSING REGISTER AND IS CHECKED FOR PROPER
                                        ;INCREMENTING.  BYTE INSTRUCTIONS ARE ALSO CHECKED.
                                        ;
                                        ;*******************************************************************************
                                        ;TEST 166       TEST ROTATE INSTRUCTIONS W/ MODE 2
                                        ;*******************************************************************************
011622                                  TS166:
011622  005000                                  CLR     R0                      ;POINT TO LOC 0
011624  012710  173737                          MOV     #173737,(R0)            ;INITIALIZE DATA
011630  000241                                  CLC                             ;CLEAR C-BIT
011632  006120                                  ROL     (R0)+                   ;TRY ROL W/ MODE 2
011634  103007                                  BCC     ROT2A                   ;CHECK C-BIT
011636  022737  167676  000000                  CMP     #167676,@#0             ;CHECK DATA
011644  001003                                  BNE     ROT2A                   ;BRANCH IF RESULT INCORRECT
011646  005300                                  DEC     R0                      ;TEST R0
```

```
011650  005300                          DEC     R0
011652  001401                          BEQ     ROT2B
011654                          ROT2A:
011654  104000                          EMT                             ;ROL W/ MODE 2 FAILED
011656  005000                  ROT2B:  CLR     R0                      ;POINT TO LOC 0
011660  012710  004040                  MOV     #4040,(R0)              ;INITIALIZE DATA
011664  000241                          CLC                             ;CLEAR C-BIT
011666  106120                          ROLB    (R0)+                   ;TRY ROLB W/ MODE 2 EVEN BYTE
011670  103406                          BCS     ROT2C                   ;CHECK C-BIT
011672  022737  004100  000000          CMP     #4100,@#0               ;CHECK DATA
011700  001002                          BNE     ROT2C                   ;BRANCH IF DATA INCORRECT
011702  005300                          DEC     R0                      ;CHECK R0
011704  001401                          BEQ     ROT2D
011706                          ROT2C:
011706  104000                          EMT                             ;ROLB W/ MODE 2 EVEN BYTE FAILED
011710  005000                  ROT2D:  CLR     R0                      ;POINT TO LOC 0
011712  012710  004040                  MOV     #4040,(R0)              ;INITIALIZE DATA
011716  005200                          INC     R0                      ;POINT TO ODD BYTE OF DATA
011720  000261                          SEC                             ;SET C-BIT
011722  106120                          ROLB    (R0)+                   ;TRY ROL W/ MODE 2 ODD BYTE
011724  103407                          BCS     ROT2E                   ;CHECK C-BIT
011726  022737  010440  000000          CMP     #10440,@#0              ;CHECK DATA
011734  001003                          BNE     ROT2E                   ;BRANCH IF DATA INCORRECT
011736  005300                          DEC     R0                      ;CHECK R0
011740  005300                          DEC     R0
011742  001401                          BEQ     TS167
011744                          ROT2E:
011744  104000                          EMT                             ;ROLB W/ MODE 2 ODD BYTE FAILED

                                ;*************************************************************************
                                ;
                                ;       THIS TEST VERIFIES MODE 3 ROTATE INSTRUCTIONS.
                                ;THIS TEST USES THE SAME PROCEDURES AS IN THE OTHER ROTATE
                                ;TESTS.  THE DATA IS STORED IN LOC. 0 AND IS ADDRESSED USING
                                ;MODE 37.  BYTE ADDRESSING IS ALSO CHECKED FOR EVEN AND ODD BYTES.
                                ;
                                ;*************************************************************************
                                ;TEST 167       TEST ROTATE INSTRUCTIONS /W MODE 3
                                ;*************************************************************************
011746                          TS167:
011746  012737  052525  000000          MOV     #52525,@#0              ;INITIALIZE DATA IN LOC 0
011754  000261                          SEC                             ;SET C-BIT
011756  006137  000000                  ROL     @#0                     ;TRO ROL W/ MODE 3
011762  103404                          BCS     ROT3A                   ;CHECK C-BIT
011764  022737  125253  000000          CMP     #125253,@#0             ;CHECK DATA
011772  001401                          BEQ     ROT3B
011774                          ROT3A:
011774  104000                          EMT                             ;ROL W/ MODE 3 FAILED
011776  012737  125252  000000  ROT3B:  MOV     #125252,@#0             ;INITIALIZE DATA
012004  000241                          CLC                             ;CLEAR C-BIT
012006  106137  000000                  ROLB    @#0                     ;TRY ROL W/ MODE 3 EVEN BYTE
012012  103004                          BCC     ROT3C                   ;CHECK C-BIT
012014  023727  000000  125124  4$:     CMP     @#0,#125124             ;CHECK DATA
012022  001401                          BEQ     ROT3D
012024                          ROT3C:
012024  104000                          EMT                             ;ROL W/ MODE 3 EVEN BYTE FAILED
```

```
012026  012737  125252  000000  ROT3D:  MOV     #125252,@#0     ;INITIALIZE DATA IN LOC. 0
012034  000261                          SEC                     ;SET C-BIT
012036  106137  000001                  ROLB    @#1             ;TRY ROL W/ MODE 3 ODD BYTE
012042  103004                          BCC     ROT3E           ;CHECK C-BIT
012044  022737  052652  000000          CMP     #052652,@#0     ;CHECK DATA
012052  001401                          BEQ     TS170
012054                          ROT3E:
012054  104000                          EMT                     ;ROL W/ MODE 3 ODD BYTE FAILED

                                ;*******************************************************************************
                                ;
                                ;       THIS TEST VERIFIES MODE 4 ROTATE INSTRUCTIONS.  THE DATA IS
                                ;STORED IN LOC. 0.  R0 IS SET TO 2 AND THE CARRY IS SET.  AN ROL MODE 4
                                ;IS USED TO ROTATE LOCATION 0 USING R0.  THE DATA IS CHECKED
                                ;AND THE C AND V BITS ARE TESTED.  THE PROPER DECREMENTING OF
                                ;R0 IS VERIFIED.
                                ;
                                ;*******************************************************************************
                                ;TEST 170       TEST MODE 4 W/ ROTATE INSTRUCTIONS
                                ;*******************************************************************************
012056                          TS170:
012056  012737  070707  000000          MOV     #070707,@#0     ;INITIALIZE DATA IN LOC. 0
012064  012700  000002                  MOV     #2,R0           ;INITIALIZE R0 AS POINTER
012070  000261                          SEC                     ;SET C-BIT
012072  006140                          ROL     -(R0)           ;TRY ROL W/ MODE 4
012074  103406                          BCS     ROT4            ;CHECK C-BIT
012076  022737  161617  000000          CMP     #161617,@#0     ;CHECK DATA
012104  001002                          BNE     ROT4            ;BRANCH IF DATA INCORRECT
012106  005700                          TST     R0              ;CHECK MODE 4 REGISTER
012110  001401                          BEQ     TS171
012112                          ROT4:
012112  104000                          EMT                     ;ROL MODE 4 FAILED

                                ;*******************************************************************************
                                ;
                                ;       THIS TEST VERIFIES MODE 5 ROTATE INSTRUCTIONS.
                                ;THE DATA IS STORED IN A WORK LOCATION (ROTX) AT THE END OF THE
                                ;TEST CODE.  LOC. 0 IS LOADED WITH THE ADDRESS OF THE DATA (ROTX).
                                ;R0 IS SET TO 2.  THE CARRY IS CLEARED AND A MODE 5 ROL
                                ;IS EXECUTED USING R0 AS AN ADDRESSING REGISTER.  THE DATA IS
                                ;CHECKED, THE C AND V BITS TESTED, AND R0 CHECKED FOR PROPER
                                ;DECREMENTING.
                                ;
                                ;*******************************************************************************
                                ;TEST 171       TEST MODE 5 W/ ROTATE INSTRUCTIONS
                                ;*******************************************************************************
012114                          TS171:
012114  012737  012160  000000          MOV     #ROTX,@#0       ;MOVE POINTER TO LOC. 0
012122  012700  000002                  MOV     #2,R0           ;SET MODE 5 REG. TO LOC. 0
012126  012767  107070  000024          MOV     #107070,ROTX    ;INITIALIZE DATA
012134  000241                          CLC                     ;CLEAR C-BIT
012136  006150                          ROL     @-(R0)          ;TRY ROL W/ MODE 5
012140  103006                          BCC     ROT5            ;CHECK C-BIT
012142  022737  016160  012160          CMP     #016160,@#ROTX  ;CHECK DATA
012150  001002                          BNE     ROT5            ;BRANCH IF DATA INCORRECT
012152  005700                          TST     R0              ;CHECK MODE 5 REGISTER
```

```
012154  001402                           BEQ     TS172
012156                           ROT5:
012156  104000                           EMT                             ;ROL MODE 5 FAILED
012160  000000                   ROTX:   0

                                 ;*************************************************************************
                                 ;
                                 ;       THIS TEST VERIFIES MODE 6 ROTATE INSTRUCTIONS.
                                 ;IT USES THE SAME PROCEDURE AS THE ABOVE TEST EXCEPT THE
                                 ;ROTATE INSTRUCTION USES MODE 6 ADDRESSING WITH REGISTER 7.
                                 ;THE DATA IS STILL OPERATED ON IN LOC. ROTX (SEE PREVIOUS TEST).
                                 ;
                                 ;*************************************************************************
                                 ;TEST 172        TEST MODE 6 W/ ROTATE INSTRUCTIONS
                                 ;*************************************************************************
012162                           TS172:
012162  012737  125252  012160           MOV     #125252,@#ROTX  ;INITIALIZE DATA
012170  000261                           SEC                     ;SET C-BIT
012172  006167  177762                   ROL     ROTX            ;TRY ROL W/ MODE 6
012176  103004                           BCC     ROT6            ;CHECK C-BIT
012200  022737  052525  012160           CMP     #52525,@#ROTX   ;CHECK DATA
012206  001401                           BEQ     TS173
012210                           ROT6:
012210  104000                           EMT                     ;ROL W/ MODE 6 FAILED

                                 ;*************************************************************************
                                 ;
                                 ;THIS TEST VERIFIES MODE 7 ROTATE INSTRUCTIONS.
                                 ;THE DATA IS SET IN LOC. ROTX, (SEE PREVIOUS TEST).  THE ROL INSTRUCTION
                                 ;ADDRESSES IT INDIRECTLY USING MODE 7 AND INDIRECT ADDRESS LOCATION
                                 ;(ROTXAD) FOLLOWING THE TEST CODE.
                                 ;
                                 ;*************************************************************************
                                 ;TEST 173        TEST MODE 7 W/ ROTATE INSTRUCTIONS
                                 ;*************************************************************************
012212                           TS173:
012212  012737  052525  012160           MOV     #52525,@#ROTX   ;INITIALIZE DATA
012220  012737  012160  012250           MOV     #ROTX,@#ROTXAD  ;INITIALIZE ADDRESS POINTER
012226  000241                           CLC                     ;CLEAR C-BIT
012230  006177  000014                   ROL     @ROTXAD         ;TRY ROL W/ MODE 7
012234  103404                           BCS     ROT7            ;CHECK C-BIT
012236  023727  012160  125252           CMP     @#ROTX,#125252  ;CHECK DATA
012244  001402                           BEQ     TS174
012246                           ROT7:
012246  104000                           EMT                     ;ROL W/ MODE 7 FAILED
012250  000000                   ROTXAD: 0

                                 ;*************************************************************************
                                 ;
                                 ;       THIS TEST VERIFIES MODE 0 SWAB INSTRUCTION.  R0 IS SET TO
                                 ;177400.  A SWAB MODE 0 IS EXECUTED AND THE CONDITIONAL BRANCH
                                 ;IS USED TO CHECK THE SIGN OF THE RESULT.  ALSO, A COMPARISON
                                 ;IS MADE TO CHECK THE DATA RESULTS.
                                 ;
                                 ;*************************************************************************
```

```
                                        ;TEST 174       TEST MODE 0 W/ SWAB INST.
                                        ;*****************************************************************************
 012252                                 TS174:
 012252  012700  177400                         MOV     #177400,R0      ;MOVE TEST PATTERN TO R0
 012256  000300                                 SWAB    R0              ;TRY SWAB MODE 0
 012260  100401                                 BMI     SB0
 012262  104000                                 EMT                     ;SWAB DID NOT SET CC'S CORRECT
 012264  022700  000377                 SB0:    CMP     #377,R0         ;CHECK RESULT
 012270  001401                                 BEQ     TS175
 012272  104000                                 EMT                     ;RESULT OF SWAB MODE 0 FAILED

                                        ;*****************************************************************************
                                        ;
                                        ;       THIS TEST VERIFIES MODE 1 SWAB INSTRUCTION.  THE TEST
                                        ;PATTERN IS MOVED TO LOC 0.  R0 IS CLEARED AND USED AS THE ADDRESSING
                                        ;REGISTER IN THE MODE 1 SWAB.  THE DATA RESULTS ARE CHECKED WITH
                                        ;A COMPARE.
                                        ;
                                        ;*****************************************************************************
                                        ;TEST 175       TEST MODE 1 W/ SWAB INST
                                        ;*****************************************************************************
 012274                                 TS175:
 012274  012737  125652  000000                 MOV     #125652,@#0     ;MOVE TEST PATTERN TO LOC. 0
 012302  005000                                 CLR     R0              ;R0=0
 012304  000310                                 SWAB    (R0)            ;TRY SWAB MODE 1
 012306  022737  125253  000000                 CMP     #125253,@#0     ;CHECK RESULT
 012314  001401                                 BEQ     TS176
 012316  104000                                 EMT                     ;RESULT OF SWAB MODE 1 FAILED

                                        ;*****************************************************************************
                                        ;
                                        ;       THIS TEST VERIFIES MODE 2 SWAB INSTRUCTION.  THE TEST
                                        ;PATTERN IS MOVED TO LOC 0.  R0 IS CLEARED AND USED AS THE MODE
                                        ;2 ADDRESSING REGISTER.  THE RESULTS ARE CHECKED WITH A COMPARE.
                                        ;R0 IS CHECKED FOR PROPER DECREMENTING.
                                        ;
                                        ;*****************************************************************************
                                        ;TEST 176       TEST MODE 2 W/ SWAB INST
                                        ;*****************************************************************************
 012320                                 TS176:
 012320  012737  125152  000000                 MOV     #125152,@#0     ;MOVE TEST PATTERN TO LOC. 0
 012326  005000                                 CLR     R0              ;R0=0
 012330  000320                                 SWAB    (R0)+           ;TRY SWAB MODE 2
 012332  022737  065252  000000                 CMP     #65252,@#0      ;CHECK RESULT
 012340  001401                                 BEQ     SB2
 012342  104000                                 EMT                     ;RESULT OF SWAB MODE 0 FAILED
 012344  162700  000002                 SB2:    SUB     #2,R0           ;CHECK EFFECT OF REG.
 012350  001401                                 BEQ     TS177
 012352  104000                                 EMT                     ;REGISTER VALUE INCORRECT

                                        ;*****************************************************************************
                                        ;
                                        ;       THIS TEST VERIFIES MODE 3 SWAB INSTRUCTION.  THE TEST
                                        ;PATTERN IS MOVED TO LOC 0.  A MODE 3 SWAB INSTRUCTION IS EXECUTED
```

```
                                        ;USING R7 AS THE ADDRESSING REGISTER.  A COMPARE VERIFIES THE
                                        ;DATA RESULTS.
                                        ;
                                        ;*************************************************************************
                                        ;TEST 177       TEST MODE 3 W/SWAB INST.
                                        ;*************************************************************************
012354                                  TS177:
012354  012737  000377  000000                  MOV     #377,@#0        ;MOVE TEST PATTERN TO LOC. 0
012362  000337  000000                          SWAB    @#0             ;TRY SWAB W/ MODE 3
012366  022737  177400  000000                  CMP     #177400,@#0     ;CHECK RESULT
012374  001401                                  BEQ     TS200
012376  104000                                  EMT                     ;RESULT OF SWAB INCORRECT


                                        ;*************************************************************************
                                        ;
                                        ;       THIS TEST VERIFIES MODE 4 SWAB INSTRUCTIONS.  THE DATA
                                        ;IS MOVED TO LOC 0.  R0 IS SET TO 2 AND USED AS THE MODE 4 ADDRESSING
                                        ;REGISTER.  THE DATA IS CHECKED WITH A COMPARE AND R0 IS CHECKED
                                        ;FOR PROPER DECREMENTING.
                                        ;
                                        ;*************************************************************************
                                        ;TEST 200       TEST MODE 4 W/ SWAB INST
                                        ;*************************************************************************
012400                                  TS200:
012400  012737  125652  000000                  MOV     #125652,@#0     ;MOVE TEST PATTERN TO LOC. 0
012406  012700  000002                          MOV     #2,R0           ;SET UP REGISTER POINTER
012412  000340                                  SWAB    -(R0)           ;TRY SWAB MODE 4
012414  022737  125253  000000                  CMP     #125253,@#0     ;CHECK RESULT
012422  001401                                  BEQ     SB4
012424  104000                                  EMT                     ;RESULT OF SWAB INCORRECT
012426  005700                          SB4:    TST     R0              ;CHECK EFFECT ON REG.
012430  001401                                  BEQ     TS201
012432  104000                                  EMT                     ;REGISTER VALUE INCORRECT


                                        ;*************************************************************************
                                        ;
                                        ;       THIS TEST VERIFIES MODE 5 SWAB INSTRUCTION.  THE TEST USES
                                        ;TWO LOCATIONS FOLLOWING THE TEST CODE.  SB5X HOLDS THE DATA;
                                        ;SB5XAD IS A POINTER TO THE DATA LOCATION.  THE DATA IS MOVED TO
                                        ;SB5X AND R0 IS SET TO TWO PLUS THE ADDRESS OF SB5XAD.  FOLLOWING
                                        ;THE MODE 5 SWAB SB5X IS CHECKED FOR THE PROPER DATA.  R0 IS
                                        ;CHECKED TO SEE THAT IT WAS DECREMENTED PROPERLY.
                                        ;
                                        ;*************************************************************************
                                        ;TEST 201       TEST MODE 5 W/ SWAB INST.
                                        ;*************************************************************************
012434                                  TS201:
012434  012700  012476                          MOV     #SB5XAD+2,R0    ;SET UP POINTER TO WORK LOCATION
012440  012767  125125  000024                  MOV     #125125,SB5X    ;MOVE PATTERN TO WORK LOCATION
012446  000350                                  SWAB    @-(R0)          ;TRY SWAB MODE 5
012450  022767  052652  000014                  CMP     #52652,SB5X     ;CHECK RESULT
012456  001401                                  BEQ     SB5A
012460  104000                                  EMT                     ;RESULT OF SWAB INCORRECT
012462  020027  012474                  SB5A:   CMP     R0,#SB5XAD      ;CHECK RESULT OF REG.
```

```
012466  001403                          BEQ     TS202
012470                          SB5:
012470  104000                          EMT                             ;REGISTER VALUE INCORRECT
012472  000000                  SB5X:   0                               ;WORK LOCATION
012474  012472                  SB5XAD: SB5X


                                ;*****************************************************************************
                                ;
                                ;       THIS TEST VERIFIES MODE 6 SWAB INSTRUCTION.  THIS TEST
                                ;USES A WORK LOCATION (SB6X) FOLLOWING THE TEST CODE.  TEST DATA
                                ;IS LOADED INTO THE WORK LOCATION.  R0, THE ADDRESSING REGISTER
                                ;IS LOADED WITH 6 LESS THEN THE ADDRESS OF THE WORK LOCATION.
                                ;THE MODE 6 SWAB IS EXECUTED WITH A +6 OFFSET.  THE DATA IS
                                ;VERIFIED WITH A COMPARE.
                                ;
                                ;*****************************************************************************
                                ;TEST 202       TEST MODE 6 W/ SWAB INST.
                                ;*****************************************************************************
012476                          TS202:
012476  012767  125125  000022          MOV     #125125,SB6X    ;MOVE PATTERN TO WORK LOCATION
012504  012700  012520                  MOV     #SB6X-6,R0      ;MOVE OFFSET POINTER TO R0
012510  000360  000006                  SWAB    6(R0)           ;TRY SWAB W/ MODE 6
012514  022760  052652  000006          CMP     #52652,6(R0)    ;CHECK RESULT
012522  001402                          BEQ     TS203
012524                          SB6:
012524  104000                          EMT                     ;RESULT OF SWAB INCORRECT
012526  000000                  SB6X:   0                       ;WORK LOCATION


                                ;*****************************************************************************
                                ;
                                ;       THIS TEST VERIFIES MODE 7 SWAB INSTRUCTION.  THIS TEST
                                ;USES TWO LOCATIONS FOLLOWING THE TEST CODE:  A WORK LOCATION
                                ;(SB.X) AND A POINTER TO THE WORK LOCATION (SB7XAD).  DATA IS MOVED
                                ;TO THE WORK LOCATION.  R0 IS LOADED WITH 72 LESS THAN THE ADDRESS
                                ;OF THE ADDRESS POINTER.  THE DATA IS SWAB'ED USING A MODE 7
                                ;INSTRUCTION WITH AN OFFSET OF +72.  THE DATA IS VERIFIED WITH A
                                ;COMPARE.
                                ;
                                ;*****************************************************************************
                                ;TEST 203       TEST MODE 7 W/ SWAB INST.
                                ;*****************************************************************************
012530                          TS203:
012530  012767  177400  000022          MOV     #177400,SB7X    ;MOVE PATTERN TO WORK LOCATION
012536  012700  012470                  MOV     #SB7XAD-72,R0   ;MOVE OFFSET POINTER TO R0
012542  000370  000072                  SWAB    @72(R0)         ;TRY SWAB MODE 7
012546  027027  000072  000377          CMP     @72(R0),#377    ;CHECK RESULTS
012554  001403                          BEQ     TS204
012556                          SB7:
012556  104000                          EMT                     ;RESULT OF SWAB INCORRECT
012560  000000                  SB7X:   0                       ;WORK LOCATION
012562  012560                  SB7XAD: SB7X                    ;POINTER TO WORK LOCATION


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

```
                                        ;
                                        ;       THIS TEST VERIFIES ALL LEGAL MODES OF THE JMP INSTRUCTION.
                                        ;BECAUSE OF THE NATURE OF THE INSTRUCTION UNDER TEST, THIS TEST
                                        ;UTILIZES SEVERAL DIFFERENT TECHNIQUES.  THE CODE IS NOT EXECUTED
                                        ;IN A LINEAR FASHION.  THE DIFFERENT MODES ARE EXECUTED IN ORDER
                                        ;FROM 1-7; HOWEVER, THE CODE IS ARRANGED SO THAT CONTROL LEAP
                                        ;FROGS THRU THE TEST CODE.  THE ORDER OF APPEARANCE OF THE CODE
                                        ;IS:
                                        ;       JMP MODE 1
                                        ;       JMP MODE 3
                                        ;       JMP MODE 2
                                        ;       JMP MODE 4
                                        ;       JMP MODE 6
                                        ;       JMP MODE 5
                                        ;       JMP MODE 7
                                        ;AN INTERNAL SEQUENCE TEST (JMPSEQ) IS USED TO INSURE THAT THE
                                        ;JUMPS ARE OCCURRING IN THE PROGRAMMED SEQUENCE.
                                        ;       THE TEST IS MADE UP OF SEVERAL BLOCKS OF CODE.  EACH CODE
                                        ;BEGINS WITH A LABEL WHICH INDICATES THE MODE BEING EXECUTED IN
                                        ;THAT BLOCK.  A SIMPLE PROCEDURE IS FOLLOWED IN EACH BLOCK.  FOR
                                        ;EXAMPLE THE CODE BEGINNING AT JMP3 WILL FIRST COMPARE THE RESULTS
                                        ;OF THE PREVIOUS MODE 2 JUMP.  (ANY REGISTER CHANGES ARE VERIFIED
                                        ;AND THE SEQUENCE CHECK IS MADE).  THEN THE REGISTERS ARE SETUP
                                        ;FOR A MODE 3 JUMP TO THE NEXT TEST BLOCK (HERE, JMP4), THE SEQUENCE
                                        ;CHECKER IS UPDATED AND THE JUMP IS EXECUTED.
                                        ;       IF A FAILURE OCCURS, THE SEQUENCE CHECKER WILL ASSIST IN
                                        ;DETERMINING JUST WHICH MODE FAILED.  IF THE SEQUENCE IS CORRECT
                                        ;THEN THE ERROR DETECTED WAS A MODE FAILURE (E.G. FAILURE OF THE
                                        ;REGISTER TO BE INCREMENTED IN MODE 2 JUMP.)
                                        ;
                                        ;*****************************************************************************
                                        ;TEST 204       TEST THE JMP INSTRUCTION IN ALL MODES
                                        ;*****************************************************************************
012564                                  TS204:
012564  005067  000240                          CLR     JMPSEQ          ;ESTABLISH A SEQUENCE CHECKER
012570  012700  012634                          MOV     #JMP2,R0        ;SET R0=JUMP TARGET
012574  000110                                  JMP     (R0)            ;TRY JMP MODE 1
012576  022700  012600                  JMP3:   CMP     #.+2,R0         ;CHECK RESULT OF MODE 2 JUMP
012602  001401                                  BEQ     JMP3A
012604  104000                                  EMT                     ;REGISTER VALUE AFTER JMP MODE 2 INCORRECT
012606  026727  000216  000001          JMP3A:  CMP     JMPSEQ,#1       ;MAKE SURE JMPS ARE IN SEQUENCE: JMPSEQ=1?
012614  001401                                  BEQ     JMP3B
012616  104000                                  EMT                     ;SHOULD BE HERE FROM JMP MODE 2 ONLY
012620  012700  012632                  JMP3B:  MOV     #IJMP4,R0       ;POINT R0 TO INDIRECT JMP ADDR.
012624  005267  000200                          INC     JMPSEQ          ;UPDATE SEQUENCE CHECKER
012630  000130                                  JMP     @(R0)+          ;TRY JMP MODE 3
012632  012656                          IJMP4:  JMP4                    ;ADDRESS INDIRECT JUMP

012634  005767  000170                  JMP2:   TST     JMPSEQ          ;CHECK THAT JMPS ARE IN SEQUENCE: JMPSEQ=0?
012640  001401                                  BEQ     JMP2A
012642  104000                                  EMT                     ;SHOULD BE HERE FROM JMP MODE 1 ONLY
012644  005267  000160                  JMP2A:  INC     JMPSEQ          ;UPDATE SEQUENCE CHECKER
012650  012700  012576                          MOV     #JMP3,R0        ;SET R0=JUMP TARGET
012654  000120                                  JMP     (R0)+           ;TRY A JUMP MODE 2 TO "JMP3"
012656  022700  012634                  JMP4:   CMP     #IJMP4+2,R0     ;CHECK RESULT OF REGISTER IN MODF 3 JUMP
012662  001401                                  BEQ     JMP4A
```

```
012664  104000                          EMT                     ;REGISTER VALUE AFTER MODE 3 JUMP INCORRECT
012666  022767  000002  000134  JMP4A:  CMP     #2,JMPSEQ       ;CHECK JUMP SEQUENCE: JMPSEQ=2?
012674  001401                          BEQ     JMP4B
012676  104000                          EMT                     ;SHOULD BE ONLY FROM MODE 3 JUMP
012700  012700  012742          JMP4B:  MOV     #JMP5+2,R0      ;SET UP POINTER TO JUMP TARGET
012704  005267  000120                  INC     JMPSEQ          ;UPDATE SEQUENCE CHECKER
012710  000140                          JMP     -(R0)           ;TRY JUMP MODE 4 TO "JMP4"

012712  022767  000004  000110  JMP6:   CMP     #4,JMPSEQ       ;CHECK THAT JUMPS ARE IN SEQUENCE: JMPSEQ=4?
012720  001401                          BEQ     JMP6A
012722  104000                          EMT                     ;SHOULD BE HERE ONLY FROM MODE 5 JUMP
012724  012700  013364          JMP6A:  MOV     #JMP7+376,R0    ;SET UP OFFSET POINTER TO JUMP TARGET
012730  005267  000074                  INC     JMPSEQ          ;UPDATE JUMP SEQUENCE
012734  000160  177402                  JMP     -376(R0)        ;TRY MODE 6 JUMP

012740  022767  000003  000062  JMP5:   CMP     #3,JMPSEQ       ;CHECK THAT JUMPS ARE IN SEQUENCE: JMPSEQ=3?
012746  001401                          BEQ     JMP5A
012750  104000                          EMT                     ;SHOULD ONLY BE HERE FROM MODE 4 JUMP
012752  012700  012766          JMP5A:  MOV     #IJMP5+2,R0     ;SET UP POINTER TO INDIRECT JUMP ADDR.
012756  005267  000046                  INC     JMPSEQ          ;UPDATE JUMP SEQUENCE
012762  000150                          JMP     @-(R0)          ;TRY JUMP MODE 5 TO "JMP6"
012764  012712                  IJMP5:  JMP6                    ;INDIRECT ADDRESS POINTER

012766  022767  000005  000034  JMP7:   CMP     #5,JMPSEQ       ;CHECK JUMPS IN SEQUENCE: JMPSEQ=5?
012774  001401                          BEQ     JMP7A
012776  104000                          EMT                     ;SHOULD ONLY BE HERE FROM MODE 6 JUMP
013000  012700  013024          JMP7A:  MOV     #IJMP+10,R0     ;SET UP OFFSET POINTER TO INDIRECT ADDR.
013004  005267  000020                  INC     JMPSEQ          ;UPDATE JUMP SEQUENCE
013010  000170  177770                  JMP     @-10(R0)        ;TRY MODE 7 JUMP
013014  013016                  IJMP:   JMPCK                   ;INDIRECT ADDRESS

013016  026727  000006  000006  JMPCK:  CMP     JMPSEQ,#6       ;CHECK JUMPS IN SEQUENCE: JMPSEQ
013024  001402                          BEQ     TS205
013026  104000                          EMT                     ;SHOULD ONLY BE HERE FROM MODE 6 JUMP
013030  000000                  JMPSEQ: 0

                                ;*************************************************************************
                                ;
                                ;       THIS TEST VERIFIES ALL LEGAL MODES OF THE JSR INSTRUCTION.
                                ;THE CONCEPT OF LEAP FROGGING AND SEQUENCE CHECKING (JSRSEQ) IS
                                ;IDENTICAL TO THAT USED IN JMP TEST (SEE PREVIOUS TEST).  EACH
                                ;BLOCK OF CODE VERIFIES THE PREVIOUS JSR BY CHECKING THE SEQUENCE,
                                ;CHECKING THAT THE PC WAS SAVED IN THE SPECIFIED REGISTER, CHECKING
                                ;THAT THE SP WAS DECREMENTED, CHECKING THAT THE REGISTER WAS
                                ;SAVED ON THE STACK, AND FINALLY CHECKING THAT ANY MODE ADDRESS
                                ;REGISTER ALTERATIONS (E.G. INCREMENT REGISTER IN MODE 2) WERE
                                ;SUCCESSFUL.  R1 IS USED AS THE REGISTER IN ALL JSR INSTRUCTIONS.
                                ;       IF A FAILURE OCCURS, THE SEQUENCE CHECKER WILL ASSIST IN
                                ;DETERMINING JUST WHICH MODE FAILED.  IF THE SEQUENCE IS CORRECT
                                ;THEN THE ERROR DETECTED WAS A FUNCTIONAL FAILURE (E.G., INCORRECT
                                ;REGISTER SAVED).
                                ;
                                ;*************************************************************************
                                ;TEST 205       TEST JSR INSTRUCTION W/ ALL MODES
                                ;*************************************************************************
013032                          TS205:
```

```
013032  000402                          BR      JSR1
013034  000137  013416          JSR0:   JMP     @#JSRCK1

013040  012706  001000          JSR1:   MOV     #STBOT,R6       ;SET STACK POINTER
013044  012700  013136                  MOV     #JSR2,R0        ;SET TARGET ADDRESS
013050  005037  013376                  CLR     @#JSRSEQ        ;INITIALIZE SEQUENCE CHECKER
013054  005001                          CLR     R1              ;INITIALIZE R1
013056  005101                          COM     R1
013060  004110                          JSR     R1,(R0)         ;TRY JSR MODE 1
                                                                ; TO SCOPE: REPLACE THE MOVE INSTRUCTION   <====
                                                                ;           FOLLOWING W/ 774               <====
013062                          JSR1A:
013062  104000                          EMT                     ;JSR MODE 1 FAILED

013064  022737  000001  013376  JSR3:   CMP     #1,@# JSRSEQ    ;CHECK SEQUENCE: JSRSEQ=1?
013072  001014                          BNE     JSR3A           ;BRANCH IF OUT OF SEQUENCE
013074  020127  013212                  CMP     R1,#JSR4        ;PROPER PC SAVED?
013100  001011                          BNE     JSR3A           ;BRANCH IF PC WRONG
013102  022706  000776                  CMP     #STBOT-2,R6     ;STACK POINTER DECREMENTED?
013106  001006                          BNE     JSR3A           ;BRANCH IF SP WRONG
013110  022716  125252                  CMP     #125252,(R6)    ;REG SAVED ON STACK?
013114  001003                          BNE     JSR3A           ;BRANCH IF REG. NOT SAVED
013116  022700  013066                  CMP     #JSR3+2,R0      ;MODE 2 INCREMENT CORRECT?
013122  001401                          BEQ     JSR3B
013124                          JSR3A:
013124  104000                          EMT                     ;JSR MODE 3 MALFUNCTIONED
013126  005237  013376          JSR3B:  INC     @#JSRSEQ        ;UPDATE SEQUENCE CHECKER
013132  004137  013212                  JSR     R1,@#JSR4       ;TRY JSR MODE 4

013136  005737  013376          JSR2:   TST     @#JSRSEQ        ;CHECK SEQUENCE: JSRSEQ=0?
013142  001011                          BNE     JSR2A           ;BRANCH IF OUT OF SEQUENCE
013144  020127  013062                  CMP     R1,#JSR1A       ;PROPER PC SAVED?
013150  001006                          BNE     JSR2A           ;BRANCH IF PC WRONG
013152  022706  000776                  CMP     #STBOT-2,R6     ;R6 DECREMENT?
013156  001003                          BNE     JSR2A           ;BRANCH IF R6 IS INCORRECT
013160  021627  177777                  CMP     (R6),#-1        ;REGISTER SAVED?
013164  001401                          BEQ     JSR2B
013166                          JSR2A:
013166  104000                          EMT                     ;JSR MODE 1 MALFUNCTIONED
013170  012706  001000          JSR2B:  MOV     #STBOT,R6       ;INITIALIZE R6
013174  012701  125252                  MOV     #125252,R1      ;INITIALIZE R1
013200  005237  013376                  INC     @#JSRSEQ        ;UPDATE SEQUENCE CHECKER
013204  012700  013064                  MOV     #JSR3,R0        ;SET TARGET ADDRESS
013210  004120                          JSR     R1,(R0)+        ;TRY JSR MODE 2

013212  022737  000002  013376  JSR4:   CMP     #2,@#JSRSEQ     ;CHECK SEQUENCE: JSRSEQ=2?
013220  001003                          BNE     JSR4A           ;BRANCH IF OUT OF SEQUENCE
013222  022701  013136                  CMP     #JSR2,R1        ;PROPER PC SAVED?
013226  001401                          BEQ     JSR4B
013230                          JSR4A:
013230  104000                          EMT                     ;JSR MODE 3 MALFUNCTIONED
013232  005237  013376          JSR4B:  INC     @#JSRSEQ        ;UPDATE SEQUENCE CHECKER
013236  012700  013304                  MOV     #JSR5+2,R0      ;SET TARGET ADDRESS
013242  004140                          JSR     R1,-(R0)        ;TRY JSR MODE 4

013244  022767  000004  000124  JSR6:   CMP     #4,JSRSEQ       ;CHECK SEQUENCE: JSRSEQ=4?
```

```
013252  001006                          BNE     JSR6A           ;BRANCH IF OUT OF SEQUENCE
013254  022701  013342                  CMP     #JSR7,R1        ;PROPER PC SAVED?
013260  001003                          BNE     JSR6A           ;BRANCH IF PC WRONG
013262  022700  013372                  CMP     #JSR6AD,R0      ;MODE 5 REGISTER CORRECT?
013266  001401                          BEQ     JSR6B
013270                          JSR6A:
013270  104000                          EMT                     ;JSR MODE 5 FAILED
013272  005237  013376          JSR6B:  INC     @#JSRSEQ        ;UPDATE SEQUENCE CHECKER
013276  004167  000040                  JSR     R1,JSR7         ;TRY JSR MODE 6
013302  022767  000003  000066  JSR5:   CMP     #3,JSRSEQ       ;CHECK SEQUENCE: JSRSEQ=3?
013310  001006                          BNE     JSR5A           ;BRANCH IF OUT OF SEQUENCE
013312  022701  013244                  CMP     #JSR6,R1        ;PROPER PC SAVED?
013316  001003                          BNE     JSR5A           ;BRANCH IF PC WRONG
013320  022700  013302                  CMP     #JSR5,R0        ;CHECK MODE 4 REGISTER
013324  001401                          BEQ     JSR5B
013326                          JSR5A:
013326  104000                          EMT                     ;JSR MODE 4 MALFUNCTIONED
013330  005237  013376          JSR5B:  INC     @#JSRSEQ        ;UPDATE SEQUENCE CHECKER
013334  012700  013374                  MOV     #JSR6AD+2,R0    ;POINT R0 TO TARGET ADDRESS
013340  004150                          JSR     R1,@-(R0)       ;TRY JSR MODE 5

013342  022737  000005  013376  JSR7:   CMP     #5,@#JSRSEQ     ;CHECK SEQUENCE: JSRSEQ=5?
013350  001003                          BNE     JSR7A           ;BRANCH IF OUT OF SEQUENCE
013352  022701  013302                  CMP     #JSR5,R1        ;PROPER PC SAVED?
013356  001401                          BEQ     JSR7B
013360                          JSR7A:
013360  104000                          EMT                     ;JSR MODE 6 FAILED
013362  005237  013376          JSR7B:  INC     @#JSRSEQ        ;UPDATE SEQUENCE CHECKER
013366  004177  000002                  JSR     R1,@JSRCKAD     ;TRY JSR MODE 7

013372  013244                  JSR6AD: JSR6                    ;MODE 5 TARGET ADDRESS
013374  013400                  JSRCKAD:JSRCK                   ;MODE 7 TARGET ADDRESS
013376  000000                  JSRSEQ: 0                       ;SEQUENCE CHECKER

013400  022767  000006  177770  JSRCK:  CMP     #6,JSRSEQ       ;CHECK SEQUENCE: JSRSEQ=6?
013406  001003                          BNE     JSRCK1          ;BRANCH IF OUT OF SEQUENCE
013410  022701  013372                  CMP     #JSR6AD,R1      ;PROPER PC SAVED?
013414  001401                          BEQ     TS206
013416                          JSRCK1:
013416  104000                          EMT                     ;JSR MODE 7 MALFUNCTIONED


                                ;*************************************************************************
                                ;
                                ;       THIS TEST VERIFIES THE RTS INSTRUCTION.  THE STACK POINTER
                                ;IS INITIALIZED AND A TEST PATTERN STORED ON STACK.  R0 IS LOADED
                                ;WITH RETURN ADDRESS.  AN RTS IS EXECUTED, AND, AT THE TARGET
                                ;ADDRESS, A CHECK IS MADE THAT R0 WAS PROPERLY RESTORED FROM THE
                                ;STACK.
                                ;*************************************************************************
                                ;TEST 206       TEST RTS INSTRUCTION
                                ;*************************************************************************
013420                          TS206:
013420  012706  001000                  MOV     #STBOT,R6       ;INITIALIZE STACK POINTER
013424  012746  052525                  MOV     #52525,-(R6)    ;INITIALIZE TOP OF STACK
013430  012700  013440                  MOV     #RTS1,R0        ;INITIALIZE RETURN REGISTER
```

```
013434  000200                   RTS     R0              ;TRY RTS THROUGH R0
                                                         ; TO SCOPE: REPLACE THE MOVE INSTRUCTION   <====
                                                         ;           FOLLOWING W/ 770                <====
013436  104000                   EMT                     ;RTS FAILED
013440  022700  052525   RTS1:   CMP     #52525,R0       ;CHECK THAT R0 RESTORED FROM STACK
013444  001401                   BEQ     TS207
013446  104000                   EMT                     ;RTS MALFUNCTIONED
                         ;*************************************************************************
                         ;
                         ;       THESE NEXT FOUR TESTS VERIFY THE FUNCTIONING OF A GROUP
                         ;OF FOUR INSTRUCTIONS.  THE GROUP CONSISTS OF THE INSTRUCTIONS:
                         ;MOV, BIC, BIT, AND BIS.  THESE INSTRUCTIONS ARE SIMILAR IN THE
                         ;WAY THEY EFFECT THE C AND V BITS.  THEY ALL LEAVE THE V-BIT
                         ;CLEAR AND THE C-BIT UNAFFECTED.
                         ;       THE TEST PROCEDURE IS AS FOLLOWS:  THE N, Z, AND V BITS
                         ;ARE LOADED WITH THE COMPLEMENT OF THE EXPECTED RESULTS, THE C-BIT
                         ;IS LOADED WITH THE DESIRED RESULT.  THE INSTRUCTION IS EXECUTED
                         ;WITH DIFFERENT DATA PATTERNS AND THE RESULTS ARE VEPIFIED WITH
                         ;A SERIES OF CONDITIONAL BRANCH INSTRUCTIONS.  THE DATA IS CHOSEN
                         ;TO PRODUCT ALL POSSIBLE COMBINATIONS OF THE C AND V BITS.
                         ;
                         ;*************************************************************************
                         ;TEST 207        TEST MOV INSTRUCTION
                         ;*************************************************************************
013450                   TS207:
013450  000277                   SCC                     ;CC=0110
013452  000251                   +CLN!CLC
013454  012700  100000           MOV     #100000,R0      ;CC=1000
013460  101402                   BLOS    MOV1
013462  102401                   BVS     MOV1
013464  100401                   BMI     MOV2
013466                   MOV1:
013466  104000                   EMT                     ;MOV DID NOT SET CC'S CORRECTLY

013470  000277           MOV2:   SCC                     ;CC=1011
013472  000244                   CLZ
013474  012700  000000           MOV     #0,R0           ;CC=0101
013500  101002                   BHI     MOV3            ;C OR Z = 0?
013502  102401                   BVS     MOV3            ;V=1?
013504  100001                   BPL     TS210
013506                   MOV3:
013506  104000                   EMT                     ;MOV DID NOT SET CC'S CORRECTLY
                         ;*************************************************************************
                         ;TEST 210        TEST BIT INSTRUCTION
                         ;*************************************************************************
013510                   TS210:
013510  012700  100001           MOV     #100001,R0
013514  000277                   SCC                     ;CC=0110
013516  000251                   +CLN!CLC
013520  032700  100000           BIT     #100000,R0      ;CC=1000
013524  101402                   BLOS    BITST1
013526  102401                   BVS     BITST1
013530  100401                   BMI     BITST2
013532                   BITST1:
013532  104000                   EMT                     ;BIT DID NOT SET CC'S CORRECTLY
```

```
013534  000277                  BITST2: SCC                             ;CC=1011
013536  000244                          CLZ
013540  032700  077776                  BIT     #77776,R0               ;CC=0101
013544  101002                          BHI     BITST3
013546  102401                          BVS     BITST3
013550  100001                          BPL     TS211
013552                          BITST3:
013552  104000                          EMT                             ;BIT DID NOT SET CC'S CORRECTLY

                                ;*************************************************************************
                                ;TEST 211       TEST BIC INSTRUCTION
                                ;*************************************************************************
013554                          TS211:
013554  012700  177777                  MOV     #177777,R0
013560  000277                          SCC                             ;CC=0110
013562  000251                          +CLN!CLC
013564  042700  077777                  BIC     #77777,R0               ;CC=1000
013570  101402                          BLOS    BIC1
013572  102401                          BVS     BIC1
013574  100401                          BMI     BIC2
013576                          BIC1:
013576  104000                          EMT                             ;BIC DID NOT SET CC'S CORRECTLY
013600  000277                  BIC2:   SCC                             ;CC=1011
013602  000244                          CLZ
013604  042700  100000                  BIC     #100000,R0              ;CC=0101
013610  101002                          BHI     BIC3
013612  102401                          BVS     BIC3
013614  100001                          BPL     TS212
013616                          BIC3:
013616  104000                          EMT                             ;BIC DID NOT SET CC'S CORRECTLY
                                ;*************************************************************************
                                ;TEST 212       TEST BIS INSTRUCTION
                                ;*************************************************************************
013620                          TS212:
013620  005000                          CLR     R0                      ;R0=0
013622  000277                          SCC                             ;CC=1010
013624  000251                          +CLN!CLC
013626  052700  000000                  BIS     #0,R0                   ;CC=0100  R0=0
013632  103403                          BCS     BIS1
013634  102402                          BVS     BIS1
013636  100401                          BMI     BIS1
013640  001401                          BEQ     BIS2
013642                          BIS1:
013642  104000                          EMT                             ;BIS DID NOT SET CC'S CORRECTLY
013644  000277                  BIS2:   SCC                             ;CC=0111
013646  000250                          CLN
013650  052700  177777                  BIS     #177777,R0              ;CC=1001
013654  103003                          BCC     BIS3
013656  102402                          BVS     BIS3
013660  001401                          BEQ     BIS3
013662  100401                          BMI     TS213
013664                          BIS3:
013664  104000                          EMT                             ;BIS DID NOT SET CC'S CORRECTLY

                                ;*************************************************************************
                                ;
                                ;       THESE NEXT TWO TESTS VERIFY THE FUNCTIONING OF THE INC AND
```

```
                              ;DEC INSTRUCTIONS.  THESE INSTRUCTIONS BOTH EFFECT THE C AND V
                              ;BITS THE SAME; THE C-BIT IS LEFT UNCHANGED AND THE V-BIT IS DEPENDENT
                              ;UPON THE DATA RESULTS.  THE SAME PROCEDURE IS USED.  THE CONDITION
                              ;CODE BITS ARE INITIALIZED, THE INSTRUCTION IS EXECUTED AND THE
                              ;RESULTS ARE VERIFIED WITH A SERIES OF CONDITIONAL BRANCH INSTRUCTIONS.
                              ;THIS PROCEDURE IS REPEATED WITH SEVERAL DATA PATTERNS TO PRODUCE
                              ;DIFFERENT COMBINATIONS OF THE C AND V BITS.
                              ;
                              ;*****************************************************************************
                              ;TEST 213       TEST INC INSTRUCTION
                              ;*****************************************************************************
013666                        TS213:
013666  012700  077777                MOV     #077777,R0      ;R0=077777
013672  000257                        CCC                     ;CC=0100
013674  000264                        SEZ
013676  005200                        INC     R0              ;CC=1010   R0=10000
013700  101402                        BLOS    INC1
013702  100001                        BPL     INC1
013704  102401                        BVS     INC2
013706                        INC1:
013706  104000                        EMT                     ;INC DID NOT SET CC'S CORRECTLY
013710  052700  077777        INC2:   BIS     #77777,R0       ;R0=177777
013714  000261                        SEC                     ;CC=1011
013716  000244                        CLZ
013720  005200                        INC     R0              ;CC=0101   R0=0
013722  100403                        BMI     INC3
013724  102402                        BVS     INC3
013726  103001                        BCC     INC3
013730  001401                        BEQ     INC4
013732                        INC3:
013732  104000                        EMT                     ;INC DID NOT SET CC'S CORRECTLY

013734  000277                INC4:   SCC                     ;CC=1110
013736  000241                        CLC
013740  005200                        INC     R0              ;CC=0000   R0=1
013742  101402                        BLOS    INC5
013744  100401                        BMI     INC5
013746  100001                        BPL     TS214
013750                        INC5:
013750  104000                        EMT                     ;INC DID NOT SET CC'S CORRECTLY

                              ;*****************************************************************************
                              ;TEST 214       TEST DEC INSTRUCTION
                              ;*****************************************************************************
013752                        TS214:
013752  012700  000002                MOV     #2,R0           ;R0=2
013756  000277                        SCC                     ;CC=1111
013760  005300                        DEC     R0              ;CC=0001   R0=1
013762  100403                        BMI     DEC1
013764  001402                        BEQ     DEC1
013766  102401                        BVS     DEC1
013770  103401                        BCS     DEC2
013772                        DEC1:
013772  104000                        EMT                     ;DEC DID NOT SET CC'S CORRECTLY
013774  000261                DEC2:   SEC                     ;CC=1011
013776  000244                        CLZ
```

```
014000  005300                  DEC     R0              ;CC=0101   R0=0
014002  101002                  BHI     DEC3
014004  100401                  BMI     DEC3
014006  102001                  BVC     DEC4
014010                  DEC3:
014010  104000                  EMT                     ;DEC DID NOT SET CC'S CORRECTLY
014012  000277          DEC4:   SCC                     ;CC=0110
014014  000251                  +CLN!CLC
014016  005300                  DEC     R0              ;CC=1000   R0=177777
014020  101402                  BLOS    DEC5
014022  102401                  BVS     DEC5
014024  100401                  BMI     DEC6
014026                  DEC5:
014026  104000                  EMT                     ;DEC DID NOT SET CC'S CORRECTLY
014030  042700  077777  DEC6:   BIC     #77777,R0       ;R0=100000
014034  000277                  SCC                     ;CC=0101
014036  000252                  +CLN!CLV
014040  005300                  DEC     R0              ;CC=1011   R0=77777
014042  100403                  BMI     DEC7            ;CC=0011
014044  001402                  BEQ     DEC7
014046  102001                  BVC     DEC7
014050  103401                  BCS     TS215
014052                  DEC7:
014052  104000                  EMT                     ;DEC DID NOT SET CC'S CORRECTLY


                        ;*******************************************************************************
                        ;
                        ;       THESE NEXT THREE TESTS VERIFY THE FUNCTIONING OF THE CLR,
                        ;TST, AND SWAB INSTRUCTIONS.  THESE THREE INSTRUCTIONS ALL LEAVE
                        ;THE C AND V BITS CLEARED.  AGAIN, THE CONDITION CODES ARE PRESET,
                        ;THE INSTRUCTION EXECUTED AND THE RESULTS CHECKED WITH CONDITIONAL
                        ;BRANCH INSTRUCTIONS.  THE PROCEDURE IS REPEATED TO PRODUCE OTHER
                        ;COMBINATIONS OF CONDITION CODES.
                        ;
                        ;*******************************************************************************
                        ;TEST 215       TEST CLR INSTRUCTION
                        ;*******************************************************************************
014054                  TS215:
014054  000277                  SCC                     ;CC=1011
014056  000244                  CLZ
014060  005000                  CLR     R0              ;CC=0100   R0=0
014062  100403                  BMI     CLR1
014064  102402                  BVS     CLR1
014066  103401                  BCS     CLR1
014070  001401                  BEQ     TS216
014072                  CLR1:
014072  104000                  EMT                     ;CLR DID NOT SET CC'S CORRECTLY

                        ;*******************************************************************************
                        ;TEST 216       TEST TST INSTRUCTION
                        ;*******************************************************************************
014074                  TS216:
014074  000277                  SCC                     ;CC=1011
014076  000244                  CLZ
014100  005700                  TST     R0              ;CC=0100
```

```
014102  100403                          BMI     TEST1
014104  102402                          BVS     TEST1
014106  103401                          BCS     TEST1
014110  001401                          BEQ     TEST2
014112                          TEST1:
014112  104000                          EMT                     ;TEST DID NOT SET CC'S CORRECTLY
014114  005300                  TEST2:  DEC     R0              ;MAKE R0 NEGATIVE
014116  000277                          SCC                     ;CC=0111
014120  000250                          CLN
014122  005700                          TST     R0              ;CC=1000
014124  101402                          BLOS    TEST3
014126  102401                          BVS     TEST3
014130  100401                          BMI     TS217
014132                          TEST3:
014132  104000                          EMT                     ;TEST DID NOT SET CC'S CORRECTLY
                                ;*************************************************************************
                                ;TEST 217       TEST SWAB INSTRUCTION
                                ;*************************************************************************
014134                          TS217:
014134  012700  170000                  MOV     #170000,R0      ;R0=170000
014140  000277                          SCC                     ;CC=0111
014142  000250                          CLN
014144  000300                          SWAB    R0              ;CC=1000   R0=360
014146  101402                          BLOS    SWB1
014150  102401                          BVS     SWB1
014152  100401                          BMI     SWB2
014154                          SWB1:
014154  104000                          EMT                     ;SWAB DID NOT SET CC'S CORRECTLY
014156  000277                  SWB2:   SCC                     ;CC=1011
014160  000244                          CLZ
014162  000300                          SWAB    R0              ;CC=0100   R0=170000
014164  102403                          BVS     SWB3
014166  103402                          BCS     SWB3
014170  100401                          BMI     SWB3
014172  001401                          BEQ     TS220
014174                          SWB3:
014174  104000                          EMT                     ;

                                ;*************************************************************************
                                ;
                                ;       THESE NEXT TWO TESTS VERIFY THE FUNCTIONING OF THE ADD AND
                                ;ADC INSTRUCTIONS.  BOTH OF THESE INSTRUCTIONS HANDLE THE C AND
                                ;V BITS IDENTICALLY.  THE PROCEDURE IS TO PRESET THE CONDITION
                                ;CODES, EXECUTE THE INSTRUCTION WITH A PARTICULAR SET OF DATA, AND
                                ;THEN CHECK THE RESULTS BY EXECUTING A SERIES OF CONDITIONAL
                                ;BRANCHES.  THES PROCEDURE IS REPEATED SEVERAL TIMES WITH DIFFERENT
                                ;DATA TO PRODUCE EVERY COMBINATION OF C AND V BITS.
                                ;
                                ;*************************************************************************
                                ;TEST 220       TEST ADD INSTRUCTION
                                ;*************************************************************************
014176                          TS220:
014176  012700  040000                  MOV     #40000,R0       ;R0=40000
014202  000277                          SCC                     ;CC=1111
014204  062700  030000                  ADD     #30000,R0       ;CC=0000   R0=70000
014210  101402                          BLOS    ADD1
```

```
014212  102401                          BVS     ADD1
014214  100001                          BPL     ADD2
014216                          ADD1:
014216  104000                          EMT                     ;ADC DID NOT SET CC'S CORRECTLY
014220  000264                  ADD2:   SEZ                     ;CC=0100

014222  062700  010000                  ADD     #10000,R0       ;CC=1010   40=100000
014226  101402                          BLOS    ADD3
014230  102001                          BVC     ADD3
014232  100401                          BMI     ADD4
014234                          ADD3:
014234  104000                          EMT                     ;ADD DID NOT SET CC'S CORRECTLY
014236  000257                  ADD4:   CCC                     ;CC=1000
014240  000270                          SEN
014242  062700  100000                  ADD     #100000,R0      ;CC=0111   R0=0
014246  101002                          BHI     ADD5
014250  102001                          BVC     ADD5
014252  100001                          BPL     ADD6
014254                          ADD5:
014254  104000                          EMT                     ;ADD DID NOT SET CC'S CORRECTLY
014256  062700  177777          ADD6:   ADD     #177777,R0      ;CC=1000   R0-177777
014262  101402                          BLOS    ADD7
014264  102401                          BVS     ADD7
014266  100401                          BMI     ADD8
014270                          ADD7:
014270  104000                          EMT                     ;ADD DID NOT SET CC'S CORRECTLY
014272  000277                  ADD8:   SCC                     ;CC=1010
014274  000245                          +CLC!CLZ
014276  062700  000001                  ADD     #1,R0           ;CC=0101   R=0
014302  102403                          BVS     ADD9
014304  103002                          BCC     ADD9
014306  100401                          BMI     ADD9
014310  001401                          BEQ     TS221
014312                          ADD9:
014312  104000                          EMT                     ;ADD DID NOT SET CC'S CORRECTLY

                                ;*********************************************************************
                                ;TEST 221       TEST ADC INSTRUCTION
                                ;*********************************************************************
014314                          TS221:
014314  012700  077777                  MOV     #077777,R0
014320  000277                          SCC                     ;CC=0101
014322  000252                          +CLN!CLV
014324  005500                          ADC     R0              ;CC=1010
014326  101402                          BLOS    ADC1
014330  102001                          BVC     ADC1
014332  100401                          BMI     ADC2
014334                          ADC1:
014334  104000                          EMT                     ;ADC DID NOT SET CC'S CORRECTLY
014336  052700  077777          ADC2:   BIS     #77777,R0
014342  000277                          SCC                     ;CC=1011
014344  000244                          CLZ
014346  005500                          ADC     R0              ;CC=0101   R0=0
014350  101002                          BHI     ADC3
014352  102401                          BVS     ADC3
014354  100001                          BPL     ADC4
```

```
014356                          ADC3:
014356  104000                          EMT                     ;ADC DID NOT SET CC'S CORRECTLY
014360  000277                  ADC4:   SCC
014362  000245                          +CLZ!CLC                        ;CC=1010
014364  005500                          ADC     R0              ;CC=0100
014366  102403                          BVS     ADC5
014370  103402                          BCS     ADC5
014372  100401                          BMI     ADC5
014374  001401                          BEQ     TS222
014376                          ADC5:
014376  104000                          EMT                     ;ADC DID NOT SET CC'S CORRECTLY

                                ;*******************************************************************************
                                ;
                                ;       THESE NEXT THREE TESTS VERIFY THE FUNCTIONING OF THE NEG,
                                ;CMP, AND COM INSTRUCTIONS.  EACH OF THESE INSTRUCTIONS GENERATE
                                ;THE C AND V BITS IDENTICALLY.  THE CONDITION CODES ARE PRESET,
                                ;THE INSTRUCTIONS EXECUTED, AND THE RESULTS CHECKED WITH A SERIES
                                ;OF CONDITIONAL BRANCH INSTRUCTIONS.  THIS PROCEDURE IS REPEATED
                                ;SEVERAL TIMES WITH DIFFERENT DATA IN ORDER TO GENERATE DIFFERENT
                                ;COMBINATIONS OF THE C AND V BITS.
                                ;
                                ;*******************************************************************************
                                ;TEST 222       TEST NEG INSTRUCTION
                                ;*******************************************************************************
014400                          TS222:
014400  012700  000001                  MOV     #1,R0
014404  000277                          SCC                     ;CC=0110
014406  000251                          +CLN!CLC
014410  005400                          NEG     R0              ;CC=1001   R0=177777
014412  103003                          BCC     NEG1
014414  102402                          BVS     NEG1
014416  001401                          BEQ     NEG1
014420  100401                          BMI     NEG2
014422                          NEG1:
014422  104000                          EMT                     ;NEG DID NOT SET CC'S CORRECTLY
014424  042700  077777          NEG2:   BIC     #77777,R0
014430  000257                          CCC                     ;CC=0100
014432  000264                          SEZ
014434  005400                          NEG     R0              ;CC=1011   R0=100000
014436  102003                          BVC     NEG3
014440  103002                          BCC     NEG3
014442  001401                          BEQ     NEG3
014444  100401                          BMI     NEG4
014446                          NEG3:
014446  104000                          EMT                     ;NEG DID NOT SET CC'S CORRECTLY
014450  005000                  NEG4:   CLR     R0
014452  000277                          SCC                     ;CC=1011
014454  000244                          CLZ
014456  005400                          NEG     R0              ;CC=0100   R0=0
014460  102403                          BVS     NEG5
014462  103402                          BCS     NEG5
014464  001001                          BNE     NEG5
014466  100001                          BPL     TS223
014470                          NEG5:
014470  104000                          EMT                     ;NEG DID NOT SET CC'S CORRECTLY
```

```
                                ;*************************************************************************
                                ;TEST 223        TEST CMP INSTRUCTION
                                ;*************************************************************************
014472                          TS223:
014472  012700  000005                  MOV     #5,R0
014476  000257                          CCC                             ;CC=1010
014500  000271                          +SEN!SEC
014502  022700  000005                  CMP     #5,R0                   ;CC=0101
014506  101002                          BHI     CMP1
014510  102401                          BVS     CMP1
014512  100001                          BPL     CMP2
014514                          CMP1:
014514  104000                          EMT                             ;CMP DID NOT SET CC'S CORRECTLY
014516  012700  100000          CMP2:   MOV     #100000,R0
014522  000277                          SCC                             ;CC=1101
014524  000242                          CLV
014526  020027  077777                  CMP     R0,#77777               ;CC=0010
014532  101402                          BLOS    CMP3
014534  102001                          BVC     CMP3
014536  100001                          BPL     CMP4
014540                          CMP3:
014540  104000                          EMT                             ;CMP DID NOT SET CC'S CORRECTLY
014542  052700  040000          CMP4:   BIS     #40000,R0               ;R0=140000
014546  000257                          CCC                             ;CC=0100
014550  000264                          SEZ
014552  022700  040000                  CMP     #40000,R0               ;CC=1011
014556  102003                          BVC     CMP5
014560  103002                          BCC     CMP5
014562  001401                          BEQ     CMP5
014564  100401                          BMI     CMP6
014566                          CMP5:
014566  104000                          EMT                             ;CMP DID NOT SET CC'S CORRECTLY
014570  042700  040000          CMP6:   BIC     #40000,R0
014574  000277                          SCC                             ;CC=1111
014576  022700  177777                  CMP     #-1,R0                  ;CC=0000
014602  101402                          BLOS    CMP7
014604  102401                          BVS     CMP7
014606  100001                          BPL     TS224
014610                          CMP7:
014610  104000                          EMT                             ;CMP DID NOT SET CC'S CORRECTLY

                                ;*************************************************************************
                                ;TEST 224        TEST COM INSTRUCTION
                                ;*************************************************************************
014612                          TS224:
014612  012700  177777                  MOV     #-1,R0
014616  000257                          CCC                             ;CC=1010
014620  000265                          +SEC!SEZ
014622  005100                          COM     R0                      ;CC=0101
014624  101002                          BHI     COM1
014626  102401                          BVS     COM1
014630  100001                          BPL     TS225
014632                          COM1:
014632  104000                          EMT                             ;COM DID NOT SET CC'S CORRECTLY
```

```

                                    ;*************************************************************************
                                    ;
                                    ;       THESE NEXT TWO TESTS VERIFY THE FUNCTIONING OF THE SUB
                                    ;AND SBC INSTRUCTIONS.  BOTH OF THESE INSTRUCTIONS HANDLE THE
                                    ;C AND V BITS IDENTICALLY.  THE PROCEDURE IS TO PRESET THE CONDITION
                                    ;CODES, EXECUTE THE INSTRUCTION WITH A PARTICULAR SET OF DATA, AND
                                    ;THEN CHECK THE RESULTS BY EXECUTING A SERIES OF CONDITIONAL
                                    ;BRANCHES  THIS PROCEDURE IS REPEATED SEVERAL TIMES WITH DIFFERENT
                                    ;DATA PATTERNS TO PROVIDE EVERY COMBINATION OF THE C AND V BITS.
                                    ;
                                    ;*************************************************************************
                                    ;TEST 225       TEST SUB INSTRUCTION
                                    ;*************************************************************************
014634                              TS225:
014634  012700  125252                      MOV     #125252,R0
014640  000257                              CCC                     ;CC=1010
014642  000271                              +SEN!SEC
014644  162700  125252                      SUB     #125252,R0      ;CC=0101   R0=0
014650  101002                              BHI     SUB1
014652  102401                              BVS     SUB1
014654  100001                              BPL     SUB2
014656                              SUB1:
014656  104000                              EMT                     ;SUB DID NOT SET CC'S CORRECTLY
014660  052700  100000              SUB2:   BIS     #100000,R0
014664  000277                              SCC                     ;CC=1101
014666  000242                              CLV
014670  162700  077777                      SUB     #77777,R0       ;CC=0010   R0=1
014674  101402                              BLOS    SUB3
014676  102001                              BVC     SUB3
014700  100001                              BPL     SUB4
014702                              SUB3:
014702  104000                              EMT                     ;
014704  005100                      SUB4:   COM     R0              ;R0=177777
014706  000277                              SCC                     ;CC=11111

014710  162700  100000                      SUB     #100000,R0      ;CC=0000   R0=77777
014714  101402                              BLOS    SUB5
014716  102401                              BVS     SUB5
014720  100001                              BPL     SUB6
014722                              SUB5:
014722  104000                              EMT                     ;SUB DID NOT SET CC'S CORRECTLY
014724  000257                      SUB6:   CCC                     ;CC=0100
014726  000264                              SEZ
014730  162700  140000                      SUB     #140000,R0      ;CC=1011
014734  102003                              BVC     SUB7
014736  103002                              BCC     SUB7
014740  001401                              BEQ     SUB7
014742  100401                              BMI     TS226
014744                              SUB7:
014744  104000                              EMT                     ;

                                    ;*************************************************************************
                                    ;TEST 226       TEST SBC INSTRUCTION
                                    ;*************************************************************************
014746                              TS226:
```

```
014746  012700  000001          MOV     #1,R0
014752  000277                  SCC                     ;CC=1011
014754  000244                  CLZ
014756  005600                  SBC     R0              ;CC=0100   R=0
014760  103403                  BCS     SBC1
014762  102402                  BVS     SBC1
014764  100401                  BMI     SBC1
014766  001401                  BEQ     SBC2
014770                  SBC1:
014770  104000                  EMT                     ;SBC DID NOT SET CC'S CORRECTLY
014772  000277          SBC2:   SCC                     ;CC=1010
014774  000245                  +CLZ!CLC
014776  005600                  SBC     R0              ;CC=0100   R=0
015000  103403                  BCS     SBC3
015002  102402                  BVS     SBC3
015004  100401                  BMI     SBC3
015006  001401                  BEQ     SBC4
015010                  SBC3:
015010  104000                  EMT                     ;SBC DID NOT SET CC'S CORRECTLY
015012  000277          SBC4:   SCC                     ;CC=0111
015014  000250                  CLN
015016  005600                  SBC     R0              ;CC=1001   R0=177777
015020  103003                  BCC     SBC5
015022  102402                  BVS     SBC5
015024  001401                  BEQ     SBC5
015026  100401                  BMI     SBC6
015030                  SBC5:
015030  104000                  EMT                     ;SBC DID NOT SET CC'S CORRECTLY
015032  042700  077777  SBC6:   BIC     #77777,R0       ;R0=100000
015036  000277                  SCC                     ;CC=1101
015040  000242                  CLV
015042  005600                  SBC     R0              ;CC=0010
015044  101402                  BLOS    SBC7
015046  102001                  BVC     SBC7
015050  100001                  BPL     TS227
015052                  SBC7:
015052  104000                  EMT                     ;SBC DID NOT SET CC'S CORRECTLY

                        ;*************************************************************************
                        ;
                        ;       THESE NEXT FOUR TESTS VERIFY THE FUNCTIONING OF THE ROL,
                        ;ROR, ASL AND ASR INSTRUCTIONS.  SPECIAL DATA PATTERNS ARE LOADED
                        ;AND ROTATED SEVERAL TIMES FOR EACH TEST.  THE CONDITION CODES
                        ;ARE PRESET BEFORE EACH ROTATION AND THE CONDITION CODES ARE
                        ;CHECKED AFTER EACH ROTATION.  THE FINAL CHECK IN EACH TEST IS
                        ;TO VERIFY THE COMMULATIVE DATA RESULT.  THE DATA PATTERNS HAVE
                        ;BEEN SELECTED TO PRODUCE ALL COMBINATIONS OF THE C AND V BITS.
                        ;
                        ;*************************************************************************
                        ;TEST 227       TEST ROL INSTRUCTION
                        ;*************************************************************************
015054                  TS227:
015054  012700  144000          MOV     #144000,R0      ;R0=144000
015060  000257                  CCC                     ;CC=0110
015062  000266                  +SEZ!SEV
015064  006100                  ROL     R0              ;CC=1001   R0=110000
```

```
015066  103003                  BCC     ROL1
015070  102402                  BVS     ROL1
015072  001401                  BEQ     ROL1
015074  100401                  BMI     ROL2
015076                  ROL1:
015076  104000                  EMT                     ;
015100  000277          ROL2:   SCC                     ;CC=1100
015102  000243                  +CLV!CLC
015104  006100                  ROL     R0              ;CC=0011   R0=020000
015106  103003                  BCC     ROL3
015110  102002                  BVC     ROL3
015112  001401                  BEQ     ROL3
015114  100001                  BPL     ROL4
015116                  ROL3:
015116  104000                  EMT                     ;ROL DID NOT SET CC'S CORRECTLY
015120  000277          ROL4:   SCC                     ;CC=0111
015122  000250                  CLN
015124  006100                  ROL     R0              ;CC=0000   R0=040001
015126  101402                  BLOS    ROL5
015130  102401                  BVS     ROL5
015132  100001                  BPL     ROL6
015134                  ROL5:
015134  104000                  EMT                     ;ROL DID NOT SET CC'S CORRECTLY
015136  000257          ROL6:   CCC                     ;CC=0101
015140  000265                  +SEZ!SEC
015142  006100                  ROL     R0              ;CC=1010   R0=100003
015144  101405                  BLOS    ROL7
015146  102004                  BVC     ROL7
015150  100003                  BPL     ROL7
015152  022700  100003          CMP     #100003,R0
015156  001401                  BEQ     TS230
015160                  ROL7:
015160  104000                  EMT                     ;ROL MALFUNCTIONED
                        ;*****************************************************************************
                        ;TEST 230       TEST ROR INSTRUCTION
                        ;*****************************************************************************
015162                  TS230:
015162  012700  000023          MOV     #23,R0          ;R0=23
015166  000277                  SCC                     ;CC=0111
015170  000250                  CLN
015172  006000                  ROR     R0              ;CC=1001   R0=100011
015174  102403                  BVS     ROR1
015176  103002                  BCC     ROR1
015200  001401                  BEQ     ROR1
015202  100401                  BMI     ROR2
015204                  ROR1:
015204  104000                  EMT                     ;ROR DID NOT SET CC'S CORRECTLY
015206  000257          ROR2:   CCC                     ;CC=1100
015210  000274                  +SEN!SEZ
015212  006000                  ROR     R0              ;CC=0011   R0=040004
015214  102003                  BVC     ROR3
015216  103002                  BCC     ROR3
015220  001401                  BEQ     ROR3
015222  100001                  BPL     ROR4
015224                  ROR3:
015224  104000                  EMT                     ;ROR DID NOT SET CC'S CORRECTLY
```

```
015226  000277                  ROR4:   SCC                             ;CC=1110
015230  000241                          CLC
015232  006000                          ROR     R0                      ;CC=0000   R0=020002
015234  101403                          BLOS    ROR5
015236  102402                          BVS     ROR5
015240  001401                          BEQ     ROR5
015242  100001                          BPL     ROR6
015244                          ROR5:
015244  104000                          EMT                             ;ROR DID NOT SET CC'S CORRECTLY
015246  000257                  ROR6:   CCC                             ;CC=0101
015250  000265                          +SEC!SEZ
015252  006000                          ROR     R0                      ;CC=1010   R0=110001
015254  101402                          BLOS    ROR7
015256  102001                          BVC     ROR7
015260  100401                          BMI     TS231
015262                          ROR7:
015262  104000                          EMT                             ;ROR DID NOT PRODUCE CORRECT RESULTS
                                ;*************************************************************************
                                ;TEST 231       TEST ASL INSTRUCTION
                                ;*************************************************************************
015264                          TS231:
015264  012700  144000                  MOV     #144000,R0              ;R0=14000
015270  000257                          CCC                             ;CC=0110
015272  000271                          +SEN!SEC
015274  006300                          ASL     R0                      ;CC=1001   R0=110000
015276  103003                          BCC     ASL1
015300  102402                          BVS     ASL1
015302  001401                          BEQ     ASL1
015304  100401                          BMI     ASL2
015306                          ASL1:
015306  104000                          EMT                             ;
015310  000277                  ASL2:   SCC                             ;CC=1100
015312  000243                          +CLV!CLC
015314  006300                          ASL     R0                      ;CC=0011   R0=020000
015316  103003                          BCC     ASL3
015320  102002                          BVC     ASL3
015322  001401                          BEQ     ASL3
015324  100001                          BPL     ASL4
015326                          ASL3:
015326  104000                          EMT                             ;ASL DID NOT SET CC'S CORRECTLY
015330  000277                  ASL4:   SCC                             ;CC=0111
015332  000250                          CLN
015334  006300                          ASL     R0                      ;CC=0000   R0=040000
015336  101402                          BLOS    ASL5
015340  102401                          BVS     ASL5
015342  100001                          BPL     ASL6
015344                          ASL5:
015344  104000                          EMT                             ;ASL DID NOT SET CC'S CORRECTLY
015346  000257                  ASL6:   CCC                             ;CC=0101
015350  000265                          +SEZ!SEC
015352  006300                          ASL     R0                      ;CC=1010   R0=100000
015354  103406                          BCS     ASL7
015356  001405                          BEQ     ASL7
015360  102004                          BVC     ASL7
015362  100003                          BPL     ASL7
015364  022700  100000                  CMP     #100000,R0
```

```
015370  001401                     BEQ     TS232
015372                      ASL7:
015372  104000                     EMT                     ;ASL MALFUNCTIONED
                            ;*******************************************************************************
                            ;TEST 232       TEST ASR INSTRUCTION
                            ;*******************************************************************************
015374                      TS232:
015374  012700  100023             MOV     #100023,R0      ;R0=100023
015400  000277                     SCC                     ;CC=0110
015402  000250                     CLN
015404  006200                     ASR     R0              ;CC=1001   RP=140011
015406  102403                     BVS     ASR1
015410  103002                     BCC     ASR1
015412  001401                     BEQ     ASR1
015414  100401                     BMI     ASR2
015416                      ASR1:
015416  104000                     EMT                     ;ASR DID NOT SET CC'S CORRECTLY
015420  042700  100000      ASR2:  BIC     #100000,R0      ;R0=40011
015424  000277                     SCC                     ;CC=1100
015426  000243                     +CLV!CLC
015430  006200                     ASR     R0              ;CC=0011   R0-020004
015432  102003                     BVC     ASR3
015434  103002                     BCC     ASR3
015436  001401                     BEQ     ASR3
015440  100001                     BPL     ASR4
015442                      ASR3:
015442  104000                     EMT                     ;ASR DID NOT SET CC'S CORRECTLY
015444  000277              ASR4:  SCC                     ;CC=1111

015446  006200                     ASR     R0              ;CC=0000   R0=010002
015450  101403                     BLOS    ASR5
015452  102402                     BVS     ASR5
015454  001401                     BEQ     ASR5
015456  100001                     BPL     ASR6
015460                      ASR5:
015460  104000                     EMT                     ;ASR DID NOT SET CC'S CORRECTLY
015462  052700  100000      ASR6:  BIS     #100000,R0      ;R0=110002
015466  000257                     CCC                     ;CC=0101
015470  000265                     +SEZ!SEC
015472  006200                     ASR     R0              ;C=1010    R0=144001
015474  101406                     BLOS    ASR7
015476  102005                     BVC     ASR7
015500  100004                     BPL     ASR7
015502  001403                     BEQ     ASR7
015504  022700  144001             CMP     #144001,R0      ;CHECK RESULT OF ASR'S
015510  001401                     BEQ     TS233
015512                      ASR7:
015512  104000                     EMT                     ;ASR DID NOT FUNCTION CORRECTLY


                            ;*******************************************************************************
                            ;TEST 233       TEST RORB INSTRUCTION
                            ;*******************************************************************************
015514                      TS233:
015514  112701  000004             MOVB    #4,R1   ;LOAD REGISTER
015520  000257                     CCC                     ;CLEAR ALL FLAGS
```

```
015522 106001                          RORB    R1              ;SHIFT  BYTE RIGHT
015524 106001                          RORB    R1              ;SHIFT BYTE RIGHT
015526 122701 000001                   CMPB    #1,R1           ;CHECK RESULT
015532 001401                          BEQ     RORB1
015534 104000                          EMT                     ;RORB DID NOT FUNCTION CORRECTLY
015536 106001                  RORB1:  RORB    R1              ;SHIFT BYTE RIGHT
015540 100403                          BMI     RORB2           ;CC=7?
015542 001002                          BNE     RORB2
015544 102001                          BVC     RORB2
015546 103401                          BCS     RORB3
015550                         RORB2:
015550 104000                          EMT                     ;RORB DID NOT SET CC'S CORRECTLY
015552 106001                  RORB3:  RORB    R1              ;SHIFT BYTE RIGHT
015554 100002                          BPL     RORB4           ;CC=12
015556 101401                          BLOS    RORB4
015560 102401                          BVS     RORB5
015562                         RORB4:
015562 104000                          EMT                     ;RORB DID NOT SET CC CORRECTLY
015564 122701 000200           RORB5:  CMPB    #200,R1         ;CHECK RESULT
015570 001401                          BEQ     RORB7
015572 104000                          EMT                     ;RORB DID NOT FUNCTION CORRECTLY
015574                         RORB7:
                               ;ROTATE ODD BYTE
015574 005000                          CLR     R0              ;MAKE R0 ZERO
015576 012710 025125                   MOV     #025125,(R0)    ;PUT STARTING VALUE IN LOC. 0
015602 005200                          INC     R0              ;MAKE R0 POINT TO ODD BYTE
015604 000257                          CCC                     ;CLEAR ALL CC
015606 000261                          SEC                     ;SEC CARRY BIT
015610 106010                          RORB    (R0)            ;SHIFT BYTE RIGHT
015612 100002                          BPL     RORB10          ;CC=12?
015614 101401                          BLOS    RORB10
015616 102401                          BVS     RORB11
015620                         RORB10:
015620 104000                          EMT                     ;RORB DID NOT SET CC'S CORRECTLY
015622 022737 112525 000000    RORB11: CMP     #112525,@#0     ;CHECK RESULT
015630 001401                          BEQ     RORB12
015632 104000                          EMT                     ;RORB DID NOT FUNCTION CORRECTLY
015634 106010                  RORB12: RORB    (R0)            ;SHIFT BYTE RIGHT
015636 100403                          BMI     RORB13          ;CC=3?
015640 001402                          BEQ     RORB13
015642 102001                          BVC     RORB13
015644 103401                          BCS     RORB14
015646                         RORB13:
015646 104000                          EMT                     ;RORB DID NOT SET CC CORRECTLY
015650 022737 045125 000000    RORB14: CMP     #045125,@#0     ;CHECK RESULT
015656 001401                          BEQ     TS234
015660 104000                          EMT                     ;RORB DID NOT FUNCTION CORRECTLY


                               ;*****************************************************************************
                               ;TEST 234        TEST ASLB INSTRUCTION
                               ;*****************************************************************************
015662                         TS234:
015662 112701 000040                   MOVB    #40,R1          ;LOAD REGISTER
015666 000257                          CCC                     ;CLEAR ALL CONDITION CODES
015670 106301                          ASLB    R1              ;SHIFT BYTE LEFT
```

```
015672  106301                         ASLB    R1              ;SHIFT BYTE LEFT
015674  100002                         BPL     ASLB2           ;CHECK CC=12
015676  101401                         BLOS    ASLB2
015700  102401                         BVS     ASLB3
015702                         ASLB2:
015702  104000                         EMT                     ;ASLB DID NOT SET CONDITION CODE CORRECTLY
015704  022701  000200         ASLB3:  CMP     #200,R1         ;CHECK RESULT
015710  001401                         BEQ     ASLB1
015712  104000                         EMT                     ;ASLB DID NOT FUNCTION CORRECTLY
015714  106301                 ASLB1:  ASLB    R1              ;SHIFT BYTE LEFT
015716  100403                         BMI     ASLB4           ;CHECK CC=7?
015720  001002                         BNE     ASLB4
015722  102001                         BVC     ASLB4
015724  103401                         BCS     TS235
015726                         ASLB4:
015726  104000                         EMT                     ;ASLB   DID NOT SET CC'S CORRECTLY


                               ;*******************************************************************************
                               ;TEST 235       TEST ASRB INSTRUCTION
                               ;*******************************************************************************
015730                         TS235:
015730  112701  000004                 MOVB    #4,R1           ;SET UP STARTING DATA
015734  000257                         CCC                     ;CLEAR ALL CONDITION CODES
015736  106201                         ASRB    R1              ;SKIFT BYTE RIGHT
015740  106201                         ASRB    R1              ;SHIFT BYTE RIGHT
015742  122701  000001                 CMPB    #1,R1           ;CHECK DATA
015746  001401                         BEQ     ASRB1
015750  104000                         EMT                     ;ASRB DID NOT SHIFT DATA CORRECTLY
015752  106201                 ASRB1:  ASRB    R1              ;SHIFT BYTE RIGHT
015754  100403                         BMI     ASRB2           ;CHECK CONDITION CODE = 7?
015756  001002                         BNE     ASRB2
015760  102001                         BVC     ASRB2
015762  103401                         BCS     ASRB3
015764                         ASRB2:
015764  104000                         EMT                     ;ASRB DID NOT SET CC'S CORRECTLY
015766  106201                 ASRB3:  ASRB    R1              ;SHIFT BYTE RIGHT
015770  103401                         BCS     ASRB4           ;CHECK CC=4
015772  001401                         BEQ     ASRB5
015774                         ASRB4:
015774  104000                         EMT                     ;ASRB DID NOT SET CC'S CORRECTLY
015776  112701  000202         ASRB5:  MOVB    #202,R1         ;PUT STARTING DATA IN REGISTER
016002  106201                         ASRB    R1              ;SHIFT BYTE RIGHT
016004  106201                         ASRB    R1              ;SHIFT BYTE RIGHT
016006  100003                         BPL     ASRB6           ;CHECK CC'S =11?
016010  001402                         BEQ     ASRB6
016012  102401                         BVS     ASRB6
016014  103401                         BCS     ASRB7
016016                         ASRB6:
016016  104000                         EMT                     ;ASRB DID NOT SET CC'S CORRECTLY
016020  122701  000340         ASRB7:  CMPB    #340,R1         ;CHECK RESULT
016024  001401                         BEQ     TS236
016026  104000                         EMT                     ;ASRB DID NOT SHIFT DATA CORRECTLY

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

```
                                  ;       THIS TEST VERIFIES THE SXT INSTRUCTION. CONDITION  CODES
                                  ;ARE PRESET IN EACH OF THE TWO POSSIBLE CASES. WITH THE N-BIT SET,
                                  ;THE TEST CHECKS FOR ALL ONES IN THE DESTINATION. WITH THE N-BIT
                                  ;CLEAR, THE DESTINATION SHOULD CONTAIN ALL ZEROES. THE DATA
                                  ;IS VERIFIED BY CONDITIONAL BRANCHES.
                                  ;
                                  ;**************************************************************************
                                  ;TEST 236       TEST THE SXT INSTRUCTION
                                  ;**************************************************************************
016030                            TS236:
016030  005000                            CLR     R0
016032  000277                            SCC                     ;SET CC=1011
016034  000244                            CLZ
016036  006700                            SXT     R0              ;TRY SXT
016040  100006                            BPL     SXT0            ;TEST CC=1001
016042  001405                            BEQ     SXT0
016044  102404                            BVS     SXT0
016046  103003                            BCC     SXT0
016050  022700  177777                    CMP     #-1,R0          ;CHECK DATA RESULT
016054  001401                            BEQ     SXT1
016056                            SXT0:
016056  104000                            EMT                     ;RESULTS OF SXT INCORRECT
016060  005000                    SXT1:   CLR     R0              ;R0=0
016062  005010                            CLR     (R0)            ;LOC. 0=0
016064  005110                            COM     (R0)            ;LOC. 0=177777
016066  000257                            CCC                     ;SET CC=0110
016070  000266                            +SEZ!SEV
016072  006710                            SXT     (R0)
016074  001005                            BNE     SXT2            ;TEST CC=0100
016076  103404                            BCS     SXT2
016100  102403                            BVS     SXT2
016102  100402                            BMI     SXT2
016104  005710                            TST     (R0)
016106  001401                            BEQ     TS237
016110                            SXT2:
016110  104000                            EMT                     ;RESULTS OF SXT INCORRECT
                                  ;**************************************************************************
                                  ;
                                  ;       THIS TEST VERIFIES THE XOR INSTRUCTION. UNIQUE PATTERNS
                                  ;OF ONES AND ZEROES ARE MOVED TO DATA REGISTERS R0 AND R1.
                                  ;AFTER THE FIRST XOR INSTRUCTION R0=36146. AN XOR IS THEN
                                  ;EXECUTED WITH THIS NEW VALUE AND THE CONTENTS OF R1 TO
                                  ;REPRODUCE THE ORIGINAL VALUE IF R0=31525.
                                  ;
                                  ;**************************************************************************
                                  ;TEST 237       TEST THE XOR INSTRUCTION
                                  ;**************************************************************************
016112                            TS237:
016112  012700  007463                    MOV     #7463,R0        ;SET UP R0
016116  012701  031525                    MOV     #31525,R1       ;SET UP R1
016122  000277                            SCC                     ;SET CC=1110
016124  000241                            CLC
016126  074100                            XOR     R1,R0           ;TRY XOR
016130  101406                            BLOS    XOR1            ;CC=0000?
016132  102405                            BVS     XOR1
016134  001404                            BEQ     XOR1
```

```
016136  100403                          BMI     XOR1
016140  022700  036146                  CMP     #36146,R0       ;DATA RESULT CORRECT?
016144  001401                          BEQ     XOR2
016146                          XOR1:
016146  104000                          EMT                     ;
016150  010104                  XOR2:   MOV     R1,R4
016152  000261                          SEC                     ;CC=1110
016154  000241                          CLC
016156  074400                          XOR     R4,R0           ;TRY XOR MODE 0,0
016160  101406                          BLOS    XOR3            ;CC=0000?
016162  102405                          BVS     XOR3
016164  001404                          BEQ     XOR3
016166  100403                          BMI     XOR3
016170  022700  007463                  CMP     #7463,R0
016174  001401                          BEQ     TS240
016176                          XOR3:
016176  104000                          EMT                     ;RESULT OF XOR INCORRECT
                                ;*************************************************************************************
                                ;
                                ;       THIS TEST VERIFIES THE SOB INSTRUCTION. R4 IS USED AS A
                                ;COUNTER WHILE R0 IS THE ADDRESS REGISTER. CONDITIONAL
                                ;BRANCHES ARE USED TO VERIFY PROPER TRANSFER OF CONTROL
                                ;WHILE R4 IS CHECKED TO INSURE PROPER DECREMENTING OF R0.
                                ;
                                ;*************************************************************************************
                                ;TEST 240       TEST SOB INSTRUCTION
                                ;*************************************************************************************
016200                          TS240:
016200  012700  000525                  MOV     #525,R0
016204  010004                          MOV     R0,R4
016206  000277                          SCC                     ;SET CC=1111
016210  101002                  SOB1:   BHI     SOB2            ;CC=1111?
016212  100001                          BPL     SOB2
016214  102401                          BVS     SOB3
016216                          SOB2:
016216  104000                          EMT                     ;
016220  005304                  SOB3:   DEC     R4              ;COUNT ITERATIONS
016222  000277                          SCC                     ;CC=1111
016224  077007                          SOB     R0,SOB1         ;DO SOB W/ R0
016226  101004                          BHI     SOB4            ;CHECK CC=1111
016230  100003                          BPL     SOB4
016232  102002                          BVC     SOB4
016234  005704                          TST     R4              ;ITERATION COUNT OK?
016236  001401                          BEQ     TS241
016240                          SOB4:
016240  104000                          EMT                     ;INCORRECT # OF BRANCHES OR CC'S CHANGED
                                ;*************************************************************************************
                                ;
                                ;       THIS TEST VERIFIES THE MARK INSTRUCTION. THE EFFECTS
                                ;OF THE MARK INSTRUCTION ARE SIMULATED BY THE PROGRAM INSTRUCTIONS.
                                ;THE CONTENTS OF R5 AND THE STACK POINTER ARE CHECKED AFTER EACH
                                ;OF THE TWO ROUTINES IN THE TEST.
                                ;
                                ;*************************************************************************************
                                ;TEST 241       TEST MARK INSTRUCTION
                                ;*************************************************************************************
```

```
016242                          TS241:
016242  012706  001000                  MOV     #STBOT,SP
016246  012746  125252                  MOV     #125252,-(SP)   ;PUT R5 VALUE ON STACK
016252  162706  000074                  SUB     #74,SP          ;EFFECTIVELY PUT 36 ARGUMENTS ON STACK
016256  012705  016274                  MOV     #MRK1,R5        ;SET NEW PC IN R5
016262  012746  006436                  MOV     #6436,-(SP)     ;PUT MARK 36 INST. ON STACK
016266  000277                          SCC                     ;SET CC=1111
016270  000116                          JMP     (SP)            ;XFER CONTL TO MARK 36 INST. ON STACK
016272  104000                          EMT                     ;MARK INST. SHOULD HAVE JUMPED TO MRK1
016274  101010                  MRK1:   BHI     MRK2            ;TEST CC UNAFFECTED
016276  100007                          BPL     MRK2            ;IE. CC=1111
016300  102006                          BVC     MRK2
016302  020527  125252                  CMP     R5,#125252      ;CHECK R5 RESTORED FROM STACK
016306  001003                          BNE     MRK2
016310  022706  001000                  CMP     #STBOT,R6       ;CHECK STACK POINTER READJUSTED CORRECTLY.
016314  001401                          BEQ     MRK3
016316                          MRK2:
016316  104000                          EMT                     ;RESULTS OF MARK INCORRECT
016320  012746  052525          MRK3:   MOV     #52525,-(SP)
016324  012746  006400                  MOV     #6400,-(SP)     ;PUT MARK 0 INST. ON STACK
016330  010605                          MOV     SP,R5           ;SET ADDR. OF MARK INST. IN R5
016332  004737  016342                  JSR     PC,@#MRK4       ;DO JSR
016336  000137  016346                  JMP     @#MRK5          ;
016342  000205                  MRK4:   RTS     R5              ;DO RTS WITH R5 TO MARK INST ON STACK
016344  104000                          EMT                     ;RTS,MARK SEQUENCE FAILED
016346  022706  001000          MRK5:   CMP     #STBOT,R6       ;STACK ADJUSTED CORRECTLY
016352  001003                          BNE     MRK6            ;IF NOT:  BR
016354  022705  052525                  CMP     #52525,R5       ;CHECK IF R5 RESTORED FROM STACK
016360  001401                          BEQ     TS242
016362                          MRK6:
016362  104000                          EMT                     ;RESULTS OF MARK INCORRECT
        177776                          PS=177776
                                ;*******************************************************************************
                                ;
                                ;       THESE NEXT SEVEN TESTS VERIFY THE MTPS INSTRUCTION IN ALL
                                ;       MODES. THE PSW IS DEFINED BY AN EQUATE STATEMENT BEFORE THE
                                ;       FIRST MTPS TEST. IN EACH TEST A PATTERN OF ONES AND
                                ;       ZEROES IS SET IN A DATA REGISTER AND MOVED TO THE PSW.
                                ;       THE DATA IN THE PSW, AND THE DATA REGISTER ADDRESS,
                                ;       ARE CHECKED TO VERIFY PROPER EXECUTION OF THE INSTRUCTION.
                                ;
                                ;*******************************************************************************
                                ;TEST 242       TEST MTPS INSTRUCTION
                                ;*******************************************************************************
016364                          TS242:
016364  012700  000377                  MOV     #377,R0
016370  000257                          CCC
016372  106400                          MTPS    R0
016374  022767  000357  161374          CMP     #357,PS
016402  001401                          BEQ     MTPS1
016404  104000                          EMT                     ;MTPS FAILED
016406  005000                  MTPS1:  CLR     R0
016410  005010                          CLR     (R0)
016412  000277                          SCC                     ;CC=1111
016414  106410                          MTPS    (R0)            ;TRY MTPS MODE 1
016416  100403                          BMI     MTPS1A          ;CHECK PS
```

```
016420  102402                          BVS     MTPS1A
016422  103401                          BCS     MTPS1A
016424  001001                          BNE     TS243
016426                          MTPS1A:
016426  104000                          EMT                     ;MTPS FAILED

                                ;*****************************************************************************
                                ;TEST 243       TEST MTPS MODE 2
                                ;*****************************************************************************
016430                          TS243:
016430  005000                          CLR     R0              ;R0=0
016432  012710  177777                  MOV     #-1,(R0)        ;LOC. 0=-1
016436  005037  177776                  CLR     @#PS            ;PS=0
016442  106420                          MTPS    (R0)+           ;TRY MTPS W/MODE 2
016444  022737  000357  177776          CMP     #357,@#PS       ;CHECK DATA
016452  001401                          BEQ     MTPS2
016454  104000                          EMT                     ;DEST. DATA INCORRECT
016456  022700  000001          MTPS2:  CMP     #1,R0           ;CHECK DEST. REGISTEP.
016462  001401                          BEQ     TS244
016464  104000                          EMT                     ;DEST REGISTER NOT INCREMENTED BY 1

                                ;*****************************************************************************
                                ;TEST 244       TEST MTPS MODE 3
                                ;*****************************************************************************
016466                          TS244:
016466  012700  000402                  MOV     #402,R0         ;R0=402
016472  005010                          CLR     (R0)            ;LOC. 402=0
016474  012737  052652  000000          MOV     #52652,@#0      ;LOC. 0=52652
016502  005037  177776                  CLR     @#PS            ;PS=0
016506  106430                          MTPS    @(R0)+          ;TRY MTPS W/MODE 3
016510  022737  000252  177776          CMP     #252,@#PS       ;CHECK DEST. DATA
016516  001401                          BEQ     MTPS3
016520  104000                          EMT                     ;DEST. DATA INCORRECT
016522  022700  000404          MTPS3:  CMP     #404,R0         ;CHECK MODE 3 REGISTER.
016526  001401                          BEQ     TS245
016530  104000                          EMT                     ;MODE 3 REGISTER INCORRECT

                                ;*****************************************************************************
                                ;TEST 245       TEST MTPS MODE 4
                                ;*****************************************************************************
016532                          TS245:
016532  012700  000001                  MOV     #1,R0           ;R0=1
016536  012737  125125  000000          MOV     #125125,@#0     ;LOC. 0 = 125125
016544  005037  177776                  CLR     @#PS            ;PS=0
016550  106440                          MTPS    -(R0)           ;TRY MTPS W/MODE 4
016552  022737  000105  177776          CMP     #105,@#PS       ;CHECK DEST. DATA
016560  001401                          BEQ     MTPS4
016562  104000                          EMT                     ;DEST. DATA INCORRECT
016564  005700                  MTPS4:  TST     R0              ;CHECK MODE 4 REGISTER
016566  001401                          BEQ     TS246
016570  104000                          EMT                     ;MODE 4 REGISTER NOT DECREMENTED BY 1

                                ;*****************************************************************************
                                ;TEST 246       TEST MTPS MODE 5
                                ;*****************************************************************************
016572                          TS246:
```

```
016572  012700  000404                    MOV     #404,R0         ;R0=404
016576  012737  177400  000000            MOV     #177400,@#0     ;LOC. 0=177400
016604  000277                            SCC                     ;SET ALL COND. CODES
016606  106450                            MTPS    @-(R0)          ;TRY MTPS W/MODE 5
016610  005737  177776                    TST     @#PS            ;CHECK DEST. DATA.
016614  001401                            BEQ     MTPS5
016616  104000                            EMT                     ;DESTINATION DATA INCORRECT
016620  022700  000402           MTPS5:   CMP     #402,R0         ;CHECK MODE 5 REGISTER
016624  001401                            BEQ     TS247
016626  104000                            EMT                     ;MODE 5 REGISTER NOT DECREMENTED BY 2

                                 ;*****************************************************************************
                                 ;TEST 247        TEST MTPS MODE 6
                                 ;*****************************************************************************
016630                           TS247:
016630  012737  052652  000000            MOV     #52652,@#0      ;LOC. 0=52652
016636  012700  000406                    MOV     #406,R0         ;R0=406
016642  005037  177776                    CLR     @#PS            ;PS=0
016646  106460  177372                    MTPS    -406(R0)        ;TRY MTPS W/MODE 6
016652  022737  000252  177776            CMP     #252,@#PS       ;CHECK DEST. DATA
016660  001401                            BEQ     MTPS6
016662  104000                            EMT                     ;DEST. DATA INCORRECT
016664  022700  000406           MTPS6:   CMP     #406,R0         ;CHECK MODE 6 REGISTER
016670  001401                            BEQ     TS250
016672  104000                            EMT                     ;MODE 6 REGISTER MODIFIED

                                 ;*****************************************************************************
                                 ;TEST 250        TEST MTPS MODE 7
                                 ;*****************************************************************************
016674                           TS250:
016674  012737  052652  000000            MOV     #52652,@#0      ;LOC. 0=52652
016702  012700  000410                    MOV     #410,R0         ;R0=410
016706  005037  177776                    CLR     @#PS            ;PS=0
016712  106470  177776                    MTPS    @-2(R0)         ;TRY MTPS W/MODE 7
016716  022737  000105  177776            CMP     #105,@#PS       ;CHECK DEST. DATA
016724  001401                            BEQ     MTPS7
016726  104000                            EMT                     ;DESTINATION DATA INCORRECT
016730  022700  000410           MTPS7:   CMP     #410,R0         ;CHECK MODE 7 REGISTER
016734  001401                            BEQ     TS251
016736  104000                            EMT                     ;MODE 7 REGISTER MODIFIED

                                 ;*****************************************************************************
                                 ;
                                 ;        THESE NEXT SEVEN TESTS VERIFY THE MFPS INSTRUCTION IN ALL
                                 ;MODES. IN EACH TEST, A PATTERN OF ONES AND ZEROES IS MOVED TO THE
                                 ;PSW, AND AN MFPS INSTRUCTION MOVES THE DATA TO A LOCATION SETUP
                                 ;BY R0, EITHER DIRECTLY OR INDIRECTLY. CONDITIONAL BRANCHES ARE
                                 ;USED TO CHECK PROPER ADDRESSING AND DATA.
                                 ;
                                 ;*****************************************************************************
                                 ;TEST 251        TEST MFPS INSTRUCTION
                                 ;*****************************************************************************
016740                           TS251:
016740  012737  000377  177776            MOV     #377,@#PS
016746  106700                            MFPS    R0
016750  022700  177757                    CMP     #177757,R0
```

```
016754  001401                                  BEQ     MFPS1
016756  104000                                  EMT                     ;MFPS FAILED

016760  005000                          MFPS1:  CLR     R0
016762  012737  177777  000000                  MOV     #-1,@#0
016770  005037  177776                          CLR     @#PS
016774  106710                                  MFPS    (R0)
016776  105737  000000                          TSTB    @#0
017002  001401                                  BEQ     TS252
017004  104000                                  EMT                     ;MFPS FAILED

                                        ;*******************************************************************************
                                        ;TEST 252       TEST MFPS MODE 2
                                        ;*******************************************************************************
017006                                  TS252:
017006  005000                                  CLR     R0              ;R0=0
017010  005010                                  CLR     (R0)            ;LOC. 0=0
017012  012737  000377  177776                  MOV     #377,@#PS       ;SET PS=357
017020  106720                                  MFPS    (R0)+           ;TRY MFPS W/MODE 2
017022  103003                                  BCC     MFPS2A          ;BR TO ERROR IF C BIT CLEAR
017024  102402                                  BVS     MFPS2A          ;BR TO ERROR IF V BIT SET
017026  001401                                  BEQ     MFPS2A          ;BR TO ERROR IF Z BIT SET
017030  100401                                  BMI     MFPS2B
017032                                  MFPS2A:
017032  104000                                  EMT                     ;COND. CODES INCORRECT
017034  022737  000357  000000          MFPS2B: CMP     #357,@#0        ;CHECK DEST. DATA
017042  001401                                  BEQ     MFPS2C
017044  104000                                  EMT                     ;DEST. DATA INCORRECT
017046  022700  000001                  MFPS2C: CMP     #1,R0           ;CHECK MODE Z REGISTER
017052  001401                                  BEQ     TS253
017054  104000                                  EMT                     ;MODE 2 REGISTER NOT INCREMENTED 1

                                        ;*******************************************************************************
                                        ;TEST 253       TEST MFPS MODE 3
                                        ;*******************************************************************************
017056                                  TS253:
017056  012700  000406                          MOV     #406,R0         ;R0=406
017062  005037  000000                          CLR     @#0             ;LOC. 0=0
017066  012737  000252  177776                  MOV     #252,@#PS       ;PS=252
017074  106730                                  MFPS    @(R0)+          ;TRY MFPS WITH MODE 3
017076  103403                                  BCS     MFPS3A          ;BR TO ERROR IF C-BIT SET
017100  102402                                  BVS     MFPS3A          ;BR TO ERROR IF V-BIT SET
017102  001401                                  BEQ     MFPS3A          ;BR TO ERROR IF Z-BIT SET
017104  100401                                  BMI     MFPS3B
017106                                  MFPS3A:
017106  104000                                  EMT                     ;CONDITION CODES INCORRECT
017110  022737  125000  000000          MFPS3B: CMP     #125000,@#0     ;CHECK DEST. DATA
017116  001401                                  BEQ     MFPS3C
017120  104000                                  EMT                     ;DEST DATA INCORRECT
017122  020027  000410                  MFPS3C: CMP     R0,#410         ;CHECK MODE 3 REGISTER.
017126  001401                                  BEQ     TS254
017130  104000                                  EMT                     ;MODE 3 REGISTER NOT INCREMENTED BY 2

                                        ;*******************************************************************************
                                        ;TEST 254       TEST MFPS MODE 4
                                        ;*******************************************************************************
```

```
        017132                                 TS254:
        017132  012700  000002                         MOV     #2,R0           ;R0=2
        017136  005037  000000                         CLR     @#0             ;LOC. 0=0
        017142  012737  000125  177776                 MOV     #125,@#PS       ;PS=125
        017150  106740                                 MFPS    -(R0)           ;TRY MFPS W/MODE 4
        017152  103003                                 BCC     MFPS4A          ;BR TO ERROR IF C-BIT CLEAR
        017154  102402                                 BVS     MFPS4A          ;BR TO ERROR IF V-BIT SET
        017156  001401                                 BEQ     MFPS4A          ;BR TO ERROR IF Z-BIT SET
        017160  100001                                 BPL     MFPS4B
        017162                                 MFPS4A:
        017162  104000                                 EMT                     ;COND. CODES INCORRECT
        017164  022737  042400  000000         MFPS4B: CMP     #42400,@#0      ;CHECK DEST. DATA
        017172  001401                                 BEQ     MFPS4C
        017174  104000                                 EMT                     ;DEST. DATA INCORRECT
        017176  020027  000001                 MFPS4C: CMP     R0,#1           ;CHECK MODE 4 REGISTER
        017202  001401                                 BEQ     TS255
        017204  104000                                 EMT                     ;MODE 4 REGISTER NOT DECREMENTED BY 1

                                               ;*****************************************************************************
                                               ;TEST 255        TEST MFPS MODE 5
                                               ;*****************************************************************************
        017206                                 TS255:
        017206  012700  000410                         MOV     #410,R0         ;R0=410
        017212  012737  177777  000000                 MOV     #-1,@#0         ;LOC. 0=-1
        017220  005037  177776                         CLR     @#PS            ;PS=0
        017224  106750                                 MFPS    @-(R0)          ;TRY MFPS W/MODE 5
        017226  103403                                 BCS     MFPS5A          ;BR TO ERROR IF C-BIT SET
        017230  102402                                 BVS     MFPS5A          ;BR TO ERROR IF V-BIT SET
        017232  100401                                 BMI     MFPS5A          ;BR TO ERROR IF N-BIT SET
        017234  001401                                 BEQ     MFPS5B
        017236                                 MFPS5A:
        017236  104000                                 EMT                     ;COND. CODES INCORRECT
        017240  022737  000377  000000         MFPS5B: CMP     #377,@#0        ;CHECK DEST. DATA
        017246  001401                                 BEQ     MFPS5C
        017250  104000                                 EMT                     ;DEST DATA INCORRECT
        017252  020027  000406                 MFPS5C: CMP     R0,#406         ;CHECK MODE 5 REGISTER
        017256  001401                                 BEQ     TS256
        017260  104000                                 EMT                     ;MODE 5 REGISTER NOT DECREMENTED BY 2

                                               ; ****************************************************************************
                                               ;TEST 256        TEST MFPS MODE 6
                                               ;*****************************************************************************
        017262                                 TS256:
        017262  012700  000401                         MOV     #401,R0         ;R0=410
        017266  005037  000000                         CLR     @#0             ;LOC. 0=0
        017272  012737  000252  177776                 MOV     #252,@#PS       ;PS=252
        017300  106760  177377                         MFPS    -401(R0)        ;TRY MFPS W/MODE 6
        017304  102403                                 BVS     MFPS6A          ;BR TO ERROR IF V-BIT SET
        017306  103402                                 BCS     MFPS6A          ;BR TO ERROR IF C-BIT SET
        017310  001401                                 BEQ     MFPS6A          ;BR TO ERROR IF Z-BIT SET
        017312  100401                                 BMI     MFPS6B
        017314                                 MFPS6A:
        017314  104000                                 EMT                     ;COND. CODES INCORRECT
        017316  022737  000252  000000         MFPS6B: CMP     #252,@#0        ;CHECK DEST. DATA
        017324  001401                                 BEQ     MFPS6C
        017326  104000                                 EMT                     ;DEST. DATA INCORRECT
```

```
017330  022700  000401          MFPS6C: CMP     #401,R0         ;CHECK DEST. REGISTER
017334  001401                          BEQ     TS257
017336  104000                          EMT                     ;DEST. DATA INCORRECT

                                ;*************************************************************************
                                ;TEST 257       TEST MFPS MODE 7
                                ;*************************************************************************
017340                          TS257:
017340  012700  000777                  MOV     #777,R0         ;R0=777
017344  005037  000000                  CLR     @#0             ;LOC. 0=0
017350  012737  000125  177776          MOV     #125,@#PS       ;PS=125
017356  106770  177407                  MFPS    @-371(R0)       ;TRY MFPS W/MODE 7
017362  102403                          BVS     MFPS7A          ;BR TO ERROR IF V-BIT SET
017364  103002                          BCC     MFPS7A          ;BR TO ERROR IF C-BIT SET
017366  001401                          BEQ     MFPS7A          ;BR TO ERROR IF Z-BIT SET
017370  100001                          BPL     MFPS7B
017372                          MFPS7A:
017372  104000                          EMT                     ;CONDITION CODE INCORRECT
017374  022737  042400  000000  MFPS7B: CMP     #42400,@#0      ;CHECK DESTINATION DATA
017402  001401                          BEQ     MFPS7C
017404  104000                          EMT                     ;DEST. DATA INCORRECT
017406  022700  000777          MFPS7C: CMP     #777,R0         ;CHECK MODE 7 REGISTER
017412  001401                          BEQ     TS260
017414  104000                          EMT                     ;MODE 7 REGISTER MODIFIED

                                ;*************************************************************************
                                ;
                                ;       THIS TEST VERIFIES THAT RESET DOES NOT CLEAR THE PSW.
                                ;THE PSW IS LOADED WITH ONES, A RESET IS ISSUED, AND THE
                                ;CONTENTS OF THE PSW ARE CHECKED TO VERIFY THAT THEY HAVE NOT
                                ;CHANGED. THIS TEST IS EXECUTED ONLY ONCE EVERY 240 (DECIMAL)
                                ;ITERATIONS OF PROGRAM.
                                ;
                                ;*************************************************************************
                                ;TEST 260       TEST THAT RESET DOES NOT CLEAR PSW
                                ;*************************************************************************
017416                          TS260:
017416  032737  000001  001020          BIT     #1, @#$ENV      ;ARE WE RUNNING UNDER APT
017424  001403                          BEQ     70$             ;IF NO THEN DO TEST
017426  005737  001006                  TST     @#$PASS         ;IS THIS FIRST PASS
017432  001011                          BNE     TS261           ;IF NO THEN SHIP TO NEXT TEST
017434                          70$:
017434  012737  000357  177776          MOV     #357,@#PS               ;MOV ONES TO PSW
017442  000005                          RESET                           ;
017444  022737  000357  177776          CMP     #357,@#PS               ;PSW CORRECT?
017452  001401                          BEQ     TS261
017454  104000                          EMT                     ;RESET ALTERED PSW
017456                          REST:

                                ;*************************************************************************
                                ;
                                ;       THE FOLLOWING TEST CHECKS THE INDEPENDENT FUNCTIONING OF BASIC
                                ;DATA PATH COMPONENTS WITH USER MODE SET.
                                ;
                                ;*************************************************************************
                                ;TEST 261       TEST USER MODE R6 CAN HOLD A ONE IN EVERY POSITION
```

```
                                        ;*******************************************************************************
017456                                  TS261:
017456  052767  140000  160312                  BIS     #USRM,PS                ;SET USER MODE
017464  012706  000001                          MOV     #1,R6                   ;SET BIT0
017470  000241                                  CLC                             ;CLEAR C-BIT
017472  006106                          USP1:   ROL     R6                      ;ROTATE 1 POSITION
017474  103376                                  BCC     USP1                    ;BR IF NOT ALL DONE
017476  001404                                  BEQ     USP1A                   ;BR IF NO BITS PICKED
017500  042767  140000  160270                  BIC     #USRM,PS                ;CLEAR USER MODE
017506  104000                                  EMT                     ;USER MODE R6 PICKED A BIT
017510  042767  140000  160260          USP1A:  BIC     #USRM,PS                ;CLEAR USER MODE


                                        ;*******************************************************************************
                                        ;
                                        ;       THIS TEST CHECKS THE INDEPENDENT FUNCTIONING OF THE USER
                                        ;AND KERNEL MODE R6'S. R6 IS SETUP AND ADDRESSED IN EACH
                                        ;OF THE TWO MODES TO VERIFY THAT THE TWO R6'S ARE INDEPENDENT
                                        ;OF EACH OTHER.
                                        ;
                                        ;*******************************************************************************
                                        ;TEST 262       TEST INDEPENDENCE OF USER AND KERNEL MODE R6'S
                                        ;*******************************************************************************
017516                                  TS262:
017516  052767  140000  160252                  BIS     #USRM,PS                ;SET USER MODE
017524  012706  177777                          MOV     #-1,R6                  ;SET USER R6 TO ALL ONES
017530  022706  177777                          CMP     #-1,R6                  ;READ AND CHECK USER R6
017534  001404                                  BEQ     USP2                    ;BR IF NO ERROR
017536  042767  140000  160232                  BIC     #USRM,PS                ;CLEAR USER MODE
017544  104000                                  EMT                     ;USER R6 WILL NOT HOLD ALL ONES
017546  042767  140000  160222          USP2:   BIC     #USRM,PS                ;SET KERNEL MODE
017554  022706  177777                          CMP     #-1,R6                  ;KERNEL MODE R6 ADDR. FROM USER MODE?>>
017560  001001                                  BNE     USP3
017562  104000                                  EMT                     ;DUAL ADDRESSING ERROR USER/KERNEL R6
017564  005006                          USP3:   CLR     R6                      ;CLEAR KERNEL MODE SP
017566  052767  140000  160202                  BIS     #USRM,PS                ;SET USER MODE
017574  022706  177777                          CMP     #-1,R6                  ;CHECK USER R6 NOT ADDR. FROM KERNEL MODE
017600  042767  140000  160170                  BIC     #USRM,PS                ;CLEAR USER MODE
017606  001401                                  BEQ     USP4                    ;BR IF NO ERROR
017610  104000                                  EMT                     ;DUAL ADDRESSING ERROR OR SEQUENCE ERROR
017612  012706  001000                  USP4:   MOV     #STBOT,R6               ;RESTORE SP USER
017616  042767  140000  160152                  BIC     #USRM,PS                ;SET KERNEL MODE
017624  012706  001000                          MOV     #STBOT,R6               ;RESTORE SP KERNEL

                                        ;*******************************************************************************
                                        ;
                                        ;       THESE NEXT TWO TESTS VERIFY MFPI AND MTPI INSTRUCTIONS
                                        ;WITH R6 IN MODE 0.
                                        ;
                                        ;*******************************************************************************
                                        ;TEST 263       TEST MFPI WITH R6 IN MODE 0
                                        ;*******************************************************************************
017630                                  TS263:
017630  012706  001000                          MOV     #STBOT,R6               ;INITIALIZE KERNEL STACK POINTER
017634  012767  140000  160134                  MOV     #USRM,PS                ;SET USER MODE.PREVIOUS KERNEL
017642  012706  000600                          MOV     #USESTK,R6              ;INITIALIZE USER STACK POINTER
```

```
017646  006506                                  MFPI    R6                      ;TRY MFPI WITH MODE 0
017650  022767  140000  160120                  CMP     #140000,PS              ;CHECK PSW
017656  001404                                  BEQ     MFPI0                   ;BR IF NO ERROR
017660  042767  140000  160110                  BIC     #USRM,PS                ;CLEAR USER MODE
017666  104000                                  EMT                     ;INCORRECT PSW FROM MFPI
017670  042767  140000  160100          MFPI0:  BIC     #USRM,PS                ;CLEAR USER MODE
017676  022767  001000  160672                  CMP     #STBOT,USESTK-2         ;CHECK DATA ON STACK
017704  001401                                  BEQ     MFPI0A                  ;BR IF NO ERROR
017706  104000                                  EMT                     ;INCORRECT DATA FROM MFPI
017710                                  MFPI0A:

                                        ;*****************************************************************************
                                        ;TEST 264       TEST MTPI WITH R6 IN MODE 0
                                        ;*****************************************************************************
017710                                  TS264:
017710  005067  160062                          CLR     PS                      ;SET KERNEL MODE
017714  005006                                  CLR     R6                      ;INITIALIZE KERNEL R6
017716  012767  140000  160052                  MOV     #USRM,PS                ;SET USER MODE/PREVIOUS KERNEL
017724  012706  000600                          MOV     #USESTK,R6              ;INITIALIZE USER STACK POINTER
017730  012746  001000                          MOV     #STBOT,-(R6)            ;SET UP TARGET DATA
017734  006606                                  MTPI    R6                      ;TRY MODE 0 MTPI
017736  022767  140000  160032                  CMP     #USRM,PS                ;CHECK PSW
017744  001404                                  BEQ     MTPI0                   ;BR IF NO ERROR
017746  042767  140000  160022                  BIC     #USRM,PS                ;CLEAR USER MODE
017754  104000                                  EMT                     ;PS INCORRECT FOLLOWING MTPI
017756  005067  160014                  MTPI0:  CLR     PS                      ;SET KERNEL MODE
017762  020627  001000                          CMP     R6,#STBOT               ;CHECK TARGET DATA
017766  001401                                  BEQ     TS265
017770  104000                                  EMT                     ;DATA INCORRECT FOLLOWING MTPI



                                        ;*****************************************************************************
                                        ;
                                        ;THE FOLLOWING TEST VERIFIES THAT NO DUAL ADDRESSING OF THE GENERAL
                                        ;REGISTERS OCCURS. ALL REGISTERS ARE CLEARED, AND A UNIQUE BIT IS SET
                                        ;IN EACH. CMP INSTRUCTIONS CHECK THAT ONLY ONE BIT IS SET IN EACH
                                        ;REGISTER.
                                        ;
                                        ;*****************************************************************************
                                        ;TEST 265       DUAL REGISTER ADDRESSING TEST
                                        ;*****************************************************************************
017772                                  TS265:
017772  005000                          BITCLR: CLR     R0              ;INITIALIZE ALL REGISTERS
017774  005001                                  CLR     R1
017776  005002                                  CLR     R2
020000  005003                                  CLR     R3
020002  005004                                  CLR     R4
020004  005005                                  CLR     R5
020006  005006                                  CLR     R6
020010  052700  000001                  BITSET: BIS     #1,R0           ;SET R0=1
020014  052701  000002                          BIS     #2,R1           ;R1=2
020020  052702  000004                          BIS     #4,R2           ;R2=4
020024  052703  000010                          BIS     #10,R3          ;R3=10
020030  052704  000020                          BIS     #20,R4          ;R4=20
020034  052705  000040                          BIS     #40,R5          ;R5=40
```

```
020040  052706  000100                  BIS     #100,R6         ;R6=100
020044  022706  000100          BITCHK: CMP     #100,R6         ;TEST THAT NO DUAL ADDRESSING OCCURRED
020050  001022                          BNE     DAERR           ;BR TO ERROR HALT IF ANY OTHER BITS ARE SET
020052  022705  000040                  CMP     #40,R5
020056  001017                          BNE     DAERR
020060  022704  000020                  CMP     #20,R4
020064  001014                          BNE     DAERR
020066  022703  000010                  CMP     #10,R3
020072  001011                          BNE     DAERR
020074  022702  000004                  CMP     #4,R2
020100  001006                          BNE     DAERR
020102  022701  000002                  CMP     #2,R1
020106  001003                          BNE     DAERR
020110  022700  000001                  CMP     #1,R0
020114  001401                          BEQ     BITCON
020116                          DAERR:
020116  104000                          EMT                     ;DUAL ADDRESSING ERROR
020120  012702  001004          BITCON: MOV     #$TESTN R2      ;RESTORE POINTER

                                ;*******************************************************************************
                                ;       THIS TEST VERIFIES THAT THE UPPER BYTE OF THE PSW IS NOT AFFECTED
                                ;WHEN THE PRIORITY LEVEL OR CC'S ARE CHANGED. ALL BITS ARE
                                ;INITIALLY SET IN THE PSW, AND THE LOW BYTE IS CLEARED. A BIT
                                ;INSTRUCTION VERIFIES THE DATA.
                                ;
                                ;*******************************************************************************
                                ;TEST 266       TEST BYTE INSTRUCTION ON PSW
                                ;*******************************************************************************
020124                          TS266:
020124  052737  170357  177776          BIS     #170357,@#PS    ;SET ALL POSSIBLE BITS IN PSW
020132  105037  177776                  CLRB    @#PS            ;CLR PR LEVEL AND CC'S
020136  013700  177776                  MOV     @#PS,R0         ;COPY CONTENTS OF PSW
020142  032700  170000                  BIT     #170000,R0      ;TEST THAT UPPER BYTE IS UNAFFECTED
020146  001003                          BNE     BTCON           ;CONTINUE IF OK
020150  005037  177776          BTERR:  CLR     @#PS            ;RETURN TO KERNEL MODE
020154  104000                          EMT                     ;BYTE INSTRUCTION ALTERED PSW
020156  005037  177776          BTCON:  CLR     @#PS            ;RETURN TO KERNEL MODE

                                ;*******************************************************************************
                                ;
                                ;       THIS TEST VERIFIES THAT A JMP INSTRUCTION DOES NOT ALTER THE
                                ;CONDITION CODES IN THE PSW. THE CC'S ARE PRESET,THE JMP IS
                                ;EXECUTED, AND CONDITIONAL BRANCHES VERIFY THE STATE OF THE CC'S.
                                ;
                                ;*******************************************************************************
                                ;TEST 267       TEST THAT JMP INSTRUCTION DOES NOT AFFECT CONDITION CODES
                                ;*******************************************************************************
020162                          TS267:
020162  000277                          SCC
020164  000252                          +CLN!CLV                ;CC=0101
020166  000167  000000                  JMP     JMPT            ;JUMP TO TEST PSW
020172  100403                  JMPT:   BMI     JMPERR          ;BR TO ERROR HALT IF N-BIT IS SET
020174  001002                          BNE     JMPERR          ;BR TO ERROR HALT IF Z-BIT IS CLEAR
020176  102401                          BVS     JMPERR          ;BR TO ERROR HALT IF V-BIT IF SET
020200  103401                          BCS     TS270
020202                          JMPERR:
```

```
020202  104000                          EMT                             ;JMP INSTRUCTION AFFECTED CC'S
                                ;*******************************************************************************
                                ;
                                ;       THIS TEST VERIFIES THE SET AND CLEAR CONDITION CODE INSTRUCTIONS.
                                ;THE TEST CONSISTS OF TWO ROUTINES, ONE TO TEST ALL CLEAR CC
                                ;INSTRUCTIONS, AND THE SECOND TO TEST ALL SET CC INSTRUCTIONS. ALL
                                ;POSSIBLE COMBINATIONS OF CONDITION CODES ARE TESTED, INCLUDING NOP'S.
                                ;       TO TEST THE CLEAR CC INSTRUCTIONS, ALL CONDITION CODES ARE
                                ;INITIALLY SET. THE INSTRUCTION IS EXECUTED, AND THE PSW IS CHECKED
                                ;TO VERIFY THE PROPER COMBINATION OF CONDITION CODES.
                                ;       TO TEST THE SET CC INSTRUCTIONS, THE CONDITION CODES ARE
                                ;INITIALLY CLEARED, AND ONLY THE REQUIRED BITS ARE SET BY THE SET CC
                                ;INSTRUCTION. THE CONTENTS OF THE PSW ARE CHECKED TO VERIFY THAT
                                ;ONLY THE REQUIRED BITS WERE SET.
                                ;
                                ;*******************************************************************************
                                ;TEST 270       TEST SET CC AND CLEAR CC INSTRUCTIONS
                                ;*******************************************************************************
020204                          TS270:
020204  012767  000240  000024          MOV     #240,CC3        ;INITIALIZE CLR CC INSTRUCTION CODES
020212  012767  000017  000032          MOV     #17,CC2         ;INITIALIZE OCTAL MAP
020220  012767  000261  000074          MOV     #261,SC3        ;INITIALIZE SET CC INSTRUCTION CODES
020226  012767  000001  000102          MOV     #1,SC4          ;INITIALIZE OCTAL MAP
020234  000277                  CLRCD:  SCC                     ;SET ALL CONDITION CODES
020236  000000                  CC3:    0                       ;CONDITION CODE INSTRUCTION
020240  013704  177776                  MOV     @#PS,R4         ;COPY THE PSW
020244  042704  177760                  BIC     #177760,R4      ;ISOLATE CONDITION CODES
020250  022704                          CMP     (PC)+,R4        ;CHECK THAT PROPER CC'S WERE CLEARED
020252  000000                  CC2:    0                       ;OCTAL REPRESENTATION OF CC'S
020254  001401                          BEQ     CON1
020256  104000                          EMT                     ;CLEAR CC INSTRUCTION FAILED
020260  005367  177766          CON1:   DEC     CC2             ;SET NEXT OCTAL MAP OF CC'S
020264  005267  177746                  INC     CC3             ;GET NEXT CLEAR CC INSTRUCTION
020270  026727  177742  000257          CMP     CC3,#257        ;TEST FOR CCC INSTRUCTION
020276  003756                          BLE     CLRCD           ;GO TEST NEXT INSTRUCTION IF NOT FOUND
020300  026727  177732  000260          CMP     CC3,#260        ;CHECK FOR NOP=260
020306  001004                          BNE     SETCD           ;GO TEST SET CC INSTRUCTIONS
020310  012767  000017  177734          MOV     #17,CC2         ;SET OCTAL MAP TO TEST NOP
020316  000746                          BR      CLRCD           ;GO TEST NOP
020320  000257                  SETCD:  CCC                     ;CLEAR ALL CONDITION CODES
020322  000000                  SC3:    0                       ;CONDITION CODE INSTRUCTION
020324  013704  177776                  MOV     @#PS,R4         ;COY PSW
020330  042704  177760                  BIC     #177760,R4      ;CLEAR AWAY UNWANTED BITS
020334  022704                          CMP     (PC)+,R4        ;CHECK THAT PROPER CC'S WERE SET
020336  000000                  SC4:    0                       ;OCTAL REPRESENTATION OF CC'S
020340  001401                          BEQ     CON2
020342                          CCERR:
020342  104000                          EMT                     ;SET CC FAILED OR SEQUENCE ERROR
020344  005267  177766          CON2:   INC     SC4             ;SET NEXT OCTAL MAP
020350  005267  177746                  INC     SC3             ;PREPARE NEXT SET CC INSTRUCTION
020354  026727  177742  000277          CMP     SC3,#277        ;FINISHED?
020362  003756                          BLE     SETCD           ;BR IF NO
020364  000167  000006                  JMP     MORO            ;JUMP TO NEXT TESTS

                                ;*******************************************************************************
                                .SBTTL  TEST INSTRUCTIONS USING SAME REGISTER FOR SOURCE & DESTINATION
```

```
                                        ;
                                        ;IN AUTO INCREMENT (DECREMENT) MODES AND
                                        ;AUTO INCREMENT (DECREMENT) DEFERRED MODES,
                                        ;CONTENTS OF THE REGISTER IN USED ARE
                                        ;INCREMENTED (DECREMENTED) BY 2
                                        ;BEFORE USED AS THE SOURCE OPERAND.
                                        ;
020370  000000  000000  000000          A:      .WORD 0,0,0
020376                                  MOR0:
                                        ;*******************************************************************************
                                        ;TEST 271       TEST AUTO-INCREMENT MODE, USING R0
                                        ;*******************************************************************************
020376                                  TS271:
020376  005037  020370                          CLR     @#A             ;CLEAR LOC A
020402  012700  020370                          MOV     #A,R0           ;R0 STORES ADDR OF A
020406  060020                                  ADD     R0,(R0)+        ;CHECK THAT R0 IS INCR BY 2 BEFORE
                                                                        ;BEING USED AS THE SOURCE OPERAND
020410  022700  020372                          CMP     #A+2,R0         ;R0 INCR BY 2?
020414  001401                                  BEQ     MOR1
020416  104000                                  EMT                     ;R0 WAS NOT INCREMENTED BY 2
                                                                        ;
020420  022737  020372  020370          MOR1:   CMP     #A+2,@#A        ;CHECK CONTENT OF R0 WAS INCR BY 2 BEFORE
                                                                        ;BEING USED IN THE "ADD" INSTR
                                                                        ;LOC A CONTAINS (A+2)?
020426  001401                                  BEQ     TS272
020430  104000                                  EMT                     ;WRONG SUM IN LOC A

                                        ;*******************************************************************************
                                        ;TEST 272       AUTO-DECREMENT MODE, USING R0
                                        ;*******************************************************************************
020432                                  TS272:
020432  005037  020370                          CLR     @#A             ;CLEAR LOC A
020436  012700  020372                          MOV     #A+2,R0         ;R0 STORES ADDR OF A+2
020442  060040                                  ADD     R0,-(R0)        ;CHECK THAT R0 IS DECR BY 2 BEFORE
                                                                        ;BEING USED AS THE SOURCE OPERAND
020444  022700  020370                          CMP     #A,R0           ;R0 DECR BY 2?
020450  001401                                  BEQ     MOR2
020452  104000                                  EMT                     ;R0 WAS NOT DECREMENTED BY 2
                                                                        ;
020454  022737  020370  020370          MOR2:   CMP     #A,@#A          ;CONTENT OF R0 WAS DECR BY 2 BEFORE
                                                                        ;BEING USED IN THE "ADD" INSTR
                                                                        ;LOC A CONTAINS (R0)
020462  001401                                  BEQ     TS273
020464  104000                                  EMT                     ;WRONG SUM IN LOC A
                                                                        ;
                                        ;*******************************************************************************
                                        ;TEST 273       TEST AUTO-INCREMENT DEFERRED MODE, USING R0
                                        ;*******************************************************************************
020466                                  TS273:
020466  005037  020370                          CLR     @#A             ;CLEAR LOC A
020472  005037  020374                          CLR     @#A+4           ;CLEAR LOC A+4
020476  012737  020370  020372                  MOV     #A,@#A+2        ;STORE ADDR A IN LOC A+2
020504  012700  020372                          MOV     #A+2,R0         ;R0 STORES ADDR A+2
020510  060030                                  ADD     R0,@(R0)+       ;CHECK THAT R0 IS INCR BY 2 BEFORE
                                                                        ;BEING USED AS THE SOURCE OPERAND
020512  022700  020374                          CMP     #A+4,R0         ;R0 INCR BY 2?
```

```
020516  001401                                  BEQ     MOR3
020520  104000                                  EMT                     ;R0 WAS NOT INCREMENTED BY 2
                                                                        ;
020522  022737  020370  020372  MOR3:           CMP     #A,@#A+2        ;LOC A+2 STILL STORES ADDR A?
020530  001401                                  BEQ     MOR4
020532  104000                                  EMT                     ;LOC A+2 STORES WRONG DATA
                                                                        ;
020534  022737  020374  020370  MOR4:           CMP     #A+4,@#A        ;CHECK CONTENT OF R0 WAS INCR BY 2 BEFORE
                                                                        ;BEING USED IN THE ''ADD'' INSTR
020542  001401                                  BEQ     MOR5
020544  104000                                  EMT                     ;LOC A STORES WRONG DATA
                                                                        ;
020546  005737  020374          MOR5:           TST     @#A+4           ;LOC A+4 STILL STORES 0?
020552  001401                                  BEQ     TS274
020554  104000                                  EMT                     ;LOC A+4 DID NOT STAY CLEAR

                                ;*************************************************************************
                                ;TEST 274       TEST AUTO-DECREMENT DEFERRED, USING R0
                                ;*************************************************************************
020556                          TS274:
020556  005037  020370                          CLR     @#A             ;CLEAR LOC A
020562  005037  020374                          CLR     @#A+4           ;CLEAR LOC A+4
020566  012700  020374                          MOV     #A+4,R0         ;R0 STORES ADDR A+4
020572  012737  020370  020372                  MOV     #A,@#A+2        ;STORE ADDR A IN LOC A+2
020600  060050                                  ADD     R0,@-(R0)       ;CHECK THAT R0 IS DECR BY 2 BEFORE
                                                                        ;BEING USED AS THE SOURCE OPERAND
020602  022700  020372                          CMP     #A+2,R0         ;R0 DECREMENTED BY 2?
020606  001401                                  BEQ     MOR6
020610  104000                                  EMT                     ;R0 WAS NOT DECREMENTED BY 2
                                                                        ;
020612  022737  020372  020370  MOR6:           CMP     #A+2,@#A        ;CHECK CONTENT OF R0 WAS DECR BY 2 BEFORE
                                                                        ;BEING USED IN THE ''ADD'' INSTR
020620  001401                                  BEQ     MOR7
020622  104000                                  EMT                     ;LOC A STORES WRONG DATA

                                                                        ;
020624  022737  020370  020372  MOR7:           CMP     #A,@#A+2        ;LOC A+2 STILL STORES A?
020632  001401                                  BEQ     MOR8
020634  104000                                  EMT                     ;LOC A+2 STORES WRONG DATA
                                                                        ;
020636  005737  020374          MOR8:           TST     @#A+4           ;LOC A+4 STILL STORES 0?
020642  001401                                  BEQ     TS275
020644  104000                                  EMT                     ;LOC A+4 DID NOT STAY CLEAR
                                                                        ;

                                ;*************************************************************************
                                .SBTTL  INSTRUCTION USING PC AS SOURCE REGISTER
                                ;
                                ;IN INDEX, INDEX DEFERRED, RELATIVE, AND
                                ;RELATIVE DEFERRED MODES, DESTINATION WILL CONTAIN
                                ;THE PC COUNT OF THE CURRENT INSTRUCTION +4.
                                ;
                                ;*************************************************************************
                                ;TEST 275       TEST PC AS SOURCE IN MODE 0, USING R0
                                ;*************************************************************************
020646                          TS275:
```

```
020646  012700  177777                  MOV     #-1,R0          ;SET ALL 1 IN R0
020652  010700                  PCN01:  MOV     PC,R0           ;STORES PC IN R0
020654  022700  020654                  CMP     #PCN01+2,R0     ;R0 STORES PC+2?
020660  001401                          BEQ     TS276
020662  104000                          EMT                     ;R0 STORED WRONG VALUE

                                ;*******************************************************************************
                                ;TEST 276       TEST PC AS SOURCE IN MODE 6, USING R0
                                ;*******************************************************************************
020664                          TS276:
020664  012700  020370                  MOV #A,R0               ;R0 STORES ADDR A
020670  010760  000004          PCN2:   MOV PC,4(R0)            ;EFFECTIVE ADDR IS A+4
020674  022737  020674  020374          CMP     #PCN2+4,@#A+4   ;LOC A+4 STORES PC+4?
020702  001401                          BEQ     TS277
020704  104000                          EMT                     ;LOC A+4 STORED WRONG VALUE

                                ;*******************************************************************************
                                ;TEST 277       TEST PC AS SOURCE IN MODE 7, USING R0
                                ;*******************************************************************************
020706                          TS277:
020706  012737  020370  020374          MOV     #A,@#A+4        ;LOC A+4 STORES ADDR A
020714  012700  020370                  MOV     #A,R0           ;R0 STORES ADDR A
020720  010770  000004          PCN3:   MOV     PC,@4(R0)       ;EFFECTIVE ADDR IS A
020724  022737  020724  020370          CMP     #PCN3+4,@#A     ;LOC A STORES PC+4?
020732  001401                          BEQ     TS300
020734  104000                          EMT                     ;LOC A STORED WRONG VALUE

                                ;*******************************************************************************
                                ;TEST 300       TEST PC AS SOURCE IN RELATIVE DEFFERED MODE ,USING R0
                                ;*******************************************************************************
020736                          TS300:
020736  012737  020372  020370          MOV     #A+2,@#A        ;LOC A STORES ADDR A+2
020744  010777  177420          PCN4:   MOV     PC,@A           ;EFFECTIVE ADDR IS A+2
020750  022737  020750  020372          CMP     #PCN4+4,@#A+2   ;LOC A+2 STORES PC+4?
020756  001401                          BEQ     TS301
020760  104000                          EMT                     ;LOC A+2 STORED WRONG VALUE

                                ;*******************************************************************************
                                ;TEST 301       TEST PC AS SOURCE IN RELATIVE MODE ,USING R0
                                ;*******************************************************************************
020762                          TS301:
020762  005037  020370                  CLR     @#A             ;CLEAR A
020766  010767  177376          PCN5:   MOV     PC,A            ;EFFECTIVE ADDR IS A
020772  022737  020772  020370          CMP     #PCN5+4,@#A     ;LOC A STORES PC+4?
021000  001401                          BEQ     TS302
021002  104000                          EMT                     ;LOCATION A STORED WRONG VALUE
                                ;*******************************************************************************
                                ;THIS TESTS THE MOVE FROM PROCESSOR TYPE INSTRUCTION(MFPT)
                                ;UPON EXECUTION R0 WILL RECIEVE THE PROCESSOR MODEL CODE
                                ;WHICH IS "000003" FOR THE DCF11-AA
                                ;*******************************************************************************
                                ;*******************************************************************************
                                ;TEST 302       TEST MFPT
                                ;*******************************************************************************
021004                          TS302:
        000007                  MFPT=000007
```

```
021004  012706  001000                  MOV     #STBOT,SP       ;INITIALIZE STACK POINT IN CASE OF TRAP
021010  000007                          MFPT                    ;GET MODEL CODE.IF THIS TRAPS AN ERROR WILL BE REPORTED
021012  022700  000003                  CMP     #3,R0           ;CHECK IF CORRECT CODE RETURNED
021016  001401                          BEQ     TS303
021020  104000                          EMT                     ;WRONG CODE RETURNED

                                ;*******************************************************************************
                                .SBTTL  THE NEXT THREE TESTS EXERCISE MASKING ACTION OF MICROCODES.
                                ;*******************************************************************************
                                ;TEST 303       TEST SUB INSTRUCTION, SM=0, DM=2
                                ;*******************************************************************************
021022                          TS303:
021022  012737  052525  000000          MOV     #052525,@#0     ;SET UP LOC 0
021030  012701  050505                  MOV     #050505,R1      ;SET UP R1
021034  005000                          CLR     R0              ;CLEAR R0
021036  160120                          SUB     R1,(R0)+        ;SUBTRACTION, SM=0,DM=2
021040  022737  002020  000000          CMP     #2020,@#0       ;CHECK DIFFERENCE AT LOC 0
021046  001401                          BEQ     TS304
021050  104000                          EMT                     ;WRONG RESULT FROM SUBTRACTION

                                ;*******************************************************************************
                                ;TEST 304       TEST MFPD WITH R0, IN MODE 2
                                ;*******************************************************************************
021052                          TS304:
021052  012737  052525  000000          MOV     #052525,@#0     ;SET UP LOC 0
021060  005000                          CLR     R0              ;CLEAR R0
021062  012767  170000  156706          MOV     #170000,PS      ;SET USER MODE ON, CURRENT & PREVIOUS
021070  012706  000600                  MOV     #USESTK,R6      ;SET USER STACK POINTER
021074  106520                          MFPD    (R0)+           ;MODE 2, MFPD
021076  005067  156674                  CLR     PS              ;SET KERNEL MODE
021102  022767  052525  157466          CMP     #052525,USESTK-2        ;CHECK DATA ON STACK
021110  001401                          BEQ     TS305
021112  104000                          EMT                     ;INCORRECT DATA FROM MFPD

                                ;*******************************************************************************
                                ;TEST 305       TEST MTPD WITH R0, IN MODE 2
                                ;*******************************************************************************
021114                          TS305:
021114  012767  170000  156654          MOV     #170000,PS      ;SET USER MODE ON, CURRENT & PREVIOUS
021122  012706  000600                  MOV     #USESTK,R6      ;SET USER STACK POINTER
021126  012746  125252                  MOV     #125252,-(R6)   ;PUSH DATA IN USER STACK
021132  012737  000000  000000          MOV     #0,@#0          ;CLEAR LOC 0
021140  005000                          CLR     R0              ;CLEAR R0
021142  106620                          MTPD    (R0)+           ;MODE 2, MTPD
021144  005067  156626                  CLR     PS              ;SET KERNEL MODE
021150  022737  125252  000000          CMP     #125252,@#0     ;CHECK DATA ON LOC 0
021156  001463                          BEQ     TESTN1
021160  104000                          EMT                     ;INCORRECT DATA FROM MTPD

021162  000402                  BRTAB:  BR      .+6
021164  001002                          BNE     .+6
021166  001402                          BEQ     .+6
021170  002002                          BGE     .+6
021172  002402                          BLT     .+6
```

```
021174  003002                  BGT     .+6
021176  003402                  BLE     .+6
021200  100002                  BPL     .+6
021202  100402                  BMI     .+6
021204  101002                  BHI     .+6
021206  101402                  BLOS    .+6
021210  102002                  BVC     .+6
021212  102402                  BVS     .+6
021214  103002                  BCC     .+6                     ;SAME AS BHIS
021216  103402                  BCS     .+6                     ;SAME AS BLO

        000002          .RADIX  2
021220  177777          YNTAB:  1111111111111111                ;BR
021222  170360                  1111000011110000                ;BNE:  Z=0
021224  007417                  0000111100001111                ;BEQ:  Z=1
021226  146063                  1100110000110011                ;BGE:  N XOR V =0
021230  031714                  0011001111001100                ;BLT:  N XOR V =1
021232  140060                  1100000000110000                ;BGT:  Z+(N XOR V) =0
021234  037717                  0011111111001111                ;BLE:  Z+(N XOR V) =1

021236  177400                  1111111100000000                ;BPL:  N=0
021240  000377                  0000000011111111                ;BMI:  N=1
021242  120240                  1010000010100000                ;BHI:  C+Z=0
021244  057537                  0101111101011111                ;BLOS: C+Z=1
021246  146314                  1100110011001100                ;BVC:  V=0
021250  031463                  0011001100110011                ;BVS:  V=1
021252  125252                  1010101010101010                ;BCC:  C=0
021254  052525                  0101010101010101                ;BCS:  C=1
        000010          .RADIX  8

                        ;*************************************************************************
                        ; THE FOLLOWING ARE SPECIAL CPU TRAP
                        ;HANDLERS TO TRAP AND REPORT SPECIAL TRAPS.
                        ;
                        ;*************************************************************************

021256                  T04:
021256  104000                  EMT                             ;TRAPPED THRU LOC. 4
021260                  T010:
021260  104000                  EMT                             ;TRAPPED THRU LOC. 10
021262                  T014:
021262  104000                  EMT                             ;TRAPPED THRU LOC. 14
021264                  T020:
021264  104000                  EMT                             ;TRAPPED THRU LOC. 20
021266                  T030:
021266  104000                  EMT                             ;TRAPPED THRU LOC. 30
021270                  T034:
021270  104000                  EMT                             ;TRAPPED THRU LOC. 34
021272                  T0114:
021272  104000                  EMT                             ;TRAPPED THRU LOC. 114
021274                  T0244:
021274  104000                  EMT                             ;TRAPPED THRU LOC. 244
021276                  T0250:
021276  104000                  EMT                             ;TRAPPED THRU LOC. 250
                        .SBTTL ** STARTING OF TRAP TEST **
```

```
                                        ;SPECIAL CASE OF ODD;.EVEN .BYTE AND REGISTER 6
               000000                   HERE=0

        021300 000000                   K1:     0
        021302 000000                   K2:     0
        021304 000000                   K3:     0
        021306 000000                   K4:     0
        021310 000000                   K5:     0
        021312 000000                   K6:     0
        021314 052525                   K7:     052525
        021316 052400                   K10:    052400
        021320 000000                   K11:    0
        021322 000000                   K12:    0
        021324 000176                   SWR:    176

        021326 032737 000001 001020     TESTN1: BIT     #1,@#$ENV
        021334 001403                           BEQ     1$
        021336 012767 001022 177760             MOV     #$SWREG,SWR
        021344                          1$:
                                        ;*****************************************************************************
                                        ;TEST 306       TEST AUTO INCREMENT AND DECREMENT OF R6 FOR WORD AND BYTES
                                        ;*****************************************************************************
        021344                          TS306:
        021344 005006                           CLR     %6
        021346 112667 156426                    MOVB    (6)+,HERE       ;SIX SHOULD INCREMENT BY TWO
        021352 020627 000002                    CMP     %6,#2
        021356 001401                           BEQ     BR1
        021360 104000                           EMT                     ;R6 DID NOT AUTO INCREMENT BY TWO

        021362 012706 001000            BR1:    MOV     #1000,%6
        021366 114627 000000                    MOVB    -(6),#HERE      ;SHOULD DECREMENT BY TWO
        021372 020627 000776                    CMP     %6,#776
        021376 001401                           BEQ     BR2
        021400 104000                           EMT                     ;R6 DID NOT AUTO DECREMENT BY 2

        021402 005006                   BR2:    CLR     %6
        021404 112626                           MOVB    (6)+,(6)+       ;DOUBLES AUTO INCREMENT OF R6
        021406 020627 000004                    CMP     %6,#4
        021412 001401                           BEQ     BR3
        021414 104000                           EMT                     ;WRONG AUTO INCREMENT OF R6

        021416 005006                   BR3:    CLR     %6
        021420 005004                           CLR     %4
        021422 122624                           CMPB    (6)+,(4)+       ;TEST INCREMENT OF R6
        021424 020627 000002                    CMP     %6,#2
        021430 001401                           BEQ     BR4
        021432 104000                           EMT                     ;WRONG INCREMENT OF R6

        021434 005006                   BR4:    CLR     %6
        021436 005004                           CLR     %4
        021440 122426                           CMPB    (4)+,(6)+       ;TEST INCREMENT OF R6
        021442 020627 000002                    CMP     %6,#2
        021446 001401                           BEQ     BR5
        021450 104000                           EMT                     ;WRONG INCREMENT OF R6

        021452 005006                   BR5:    CLR     %6
```

```
021454  005004                                  CLR     %4
021456  122624                                  CMPB    (6)+,(4)+       ;TEST INCREMENT OF R4
021460  020427  000001                          CMP     %4,#1
021464  001401                                  BEQ     BR6
021466  104000                                  EMT                     ;WRONG INCREMENT OF R4
021470  005006                          BR6:    CLR     %6
021472  005004                                  CLR     %4
021474  122426                                  CMPB    (4)+,(6)+       ;TEST INCREMENT OF R6
021476  020627  000002                          CMP     %6,#2
021502  001401                                  BEQ     BR7
021504  104000                                  EMT                     ;WRONG INCREMENT OF R6

021506  005006                          BR7:    CLR     %6
021510  005004                                  CLR     %4
021512  122426                                  CMPB    (4)+,(6)+       ;TEST INCREMENT OF R4
021514  020427  000001                          CMP     %4,#1
021520  001401                                  BEQ     BR10
021522  104000                                  EMT                     ;WRONG INCREMENT OF P4

021524  012706  001000                  BR10:   MOV     #1000,%6
021530  124627  000000                          CMPB    -(6),#HERE      ;TEST DECREMENT OF R6
021534  022706  000776                          CMP     #776,%6
021540  001401                                  BEQ     TS307
021542  104000                                  EMT                     ;WRONG DECREMENT OF R6,OR WRONG $TSTNM
                                        ;*******************************************************************************
                                        ;TEST 307       TEST TRANSFER OF .BYTE USING R6
                                        ;*******************************************************************************
021544                                  TS307:
021544  012767  123456  177536                  MOV     #123456,K5
021552  012767  050505  177520                  MOV     #050505,K1
021560  012705  021300                          MOV     #K1,%5          ;%5=(050505)K1
021564  012706  021310                          MOV     #K5,%6          ;%6=(123456)K5
021570  112625                                  MOVB    (6)+,(5)+       ;LOW .BYTE OF R6 TO R5
021572  022767  050456  177500                  CMP     #050456,K1
021600  001401                                  BEQ     BR11
021602  104000                                  EMT                     ;FALSE TRANSFER OF .BYTE

021604  012767  123456  177476          BR11:   MOV     #123456,K5
021612  012767  050505  177460                  MOV     #050505,K1
021620  012705  021300                          MOV     #K1,%5          ;%5(050505)K1
021624  012706  021312                          MOV     #K6,%6          ;%6(123456)K5
021630  114625                                  MOVB    -(6),(5)+       ;LOW .BYTE OF R6 TO R5 (DECREMENT)
021632  026727  177442  050456                  CMP     K1,#050456
021640  001401                                  BEQ     BR12
021642  104000                                  EMT                     ;FALSE R6 .BYTE TRANSFER

021644  012767  123456  177426          BR12:   MOV     #123456,K1
021652  012767  050505  177430                  MOV     #050505,K5
021660  012705  021300                          MOV     #K1,%5          ;(123456)
021664  012706  021310                          MOV     #K5,%6          ;(050505)
021670  112526                                  MOVB    (5)+,(6)+       ;LOW OF R5 TO LOW OF R6
021672  022767  050456  177410                  CMP     #050456,K5
021700  001401                                  BEQ     BR13
021702  104000                                  EMT                     ;FALSE R6 .BYTE TRANSFER

021704  012767  123456  177366          BR13:   MOV     #123456,K1
```

```
021712 012767 050505 177370            MOV     #050505,K5
021720 012705 021301                   MOV     #K1+1,%5            ;123456
021724 012706 021310                   MOV     #K5,%6              ;050505
021730 112526                          MOVB    (5)+,(6)+           ;HIGH OF R5 TO LOW OF R6
021732 026727 177352 050647            CMP     K5,#050647
021740 001401                          BEQ     BR14
021742 104000                          EMT                         ;FALSE R6 .BYTE TRANSFER

021744 012767 123456 177326    BR14:   MOV     #123456,K1
021752 012767 050505 177330            MOV     #050505,K5
021760 012705 021301                   MOV     #K1+1,%5            ;R5-123456-ODD ADDRESS
021764 012706 021310                   MOV     #K5,%6              ;R6-050505--.EVEN ADDRESS
021770 112625                          MOVB    (6)+,(5)+           ;LOW OF R6 TO HIGH OF R5
021772 022767 042456 177300            CMP     #042456,K1
022000 001401                          BEQ     TS310
022002 104000                          EMT                         ;FAILED LOW OF 6 TO HIGH OF 5,OR WRONG $TSTNM
                               ;*************************************************************************
                               ;TEST 310       TEST BYTE OPERATION WITH SEQUENTIAL ODD-EVEN ADDRESS
                               ;*************************************************************************
022004                         TS310:
022004 012706 001000                   MOV     #STBOT,R6           ;RESET THE STACK POINTER
022010 126767 177300 177277            CMPB    K7,K7+1             ;SAME .WORD LOW TO HIGH
022016 001401                          BEQ     BR15
022020 104000                          EMT                         ;SHOULD COMPARE LOW TO HIGH

022022 126767 177267 177264    BR15:   CMPB    K7+1,K7             ;COMPARE ODD TO .EVEN SAME .WORD
022030 001401                          BEQ     BR16
022032 104000                          EMT                         ;ODD TO .EVEN .BYTE FAILURE

022034 126767 177257 177252    BR16:   CMPB    K10+1,K7            ;SEQUENTIAL .BYTES
022042 001401                          BEQ     BR17
022044 104000                          EMT                         ;ODD TO .EVEN FAILED

022046 126767 177244 177236    BR17:   CMPB    K10,K6
022054 001401                          BEQ     BR20
022056 104000                          EMT                         ;.EVEN TO EVEN FAILED
022060 126767 177231 177231    BR20:   CMPB    K7+1,K10+1
022066 001401                          BEQ     BR21
022070 104000                          EMT                         ;ODD TO ODD FAILED

022072 126767 177220 177217    BR21:   CMPB    K10,K10+1
022100 0010[illegible]                 BNE     BR22
022102 104000                          EMT                         ;LOW TO HIGH IN SAME .WORD FAILED

022104 126767 177207 177205    BR22:   CMPB    K10+1,K10+1
022112 001401                          BEQ     BR23
022114 104000                          EMT                         ;HIGH TO LOW IN SAME .WORD FAILED

022116 126767 177174 177171    BR23:   CMPB    K10,K7+1
022124 001001                          BNE     TS311
022126 104000                          EMT                         ;.EVEN TO ODD FAILED,OR WRONG $TSTNM


                               ;*************************************************************************
                               ;TEST 311       TEST THAT DECREMENT R6 TO A VALUE LESS THAN 400 TRAPS
                               ;*************************************************************************
```

```
022130                           TS311:
022130  013700  000142                   MOV     @#142,R0        ;SAVE LOCATIONS 142-146 FOR WHEN THE
022134  013701  000144                   MOV     @#144,R1        ;TRAP OCCURS
022140  013702  000146                   MOV     @#146,R2        ;
022144  012706  000150                   MOV     #150,%6         ;R6 = 150
022150  012767  022162  155626           MOV     #TDEC1,4        ;STACK OVERFLOW TRAP POINTER
022156  005746                           TST     -(6)            ;WITH R6 = 150 SHOULD TRAP
022160  104000                           EMT                     ;SHOULD HAVE TRAPPED,OR WRONG $TSTNM
022162                           TDEC1:

                                 ;*************************************************************************
                                 ;TEST 312        TEST FOR DECREMENT OF R6 ON OVERFLOW TRAP
                                 ;*************************************************************************
022162                           TS312:
022162  012706  000150                   MOV     #150,%6         ;R6 = 150
022166  012767  022176  155610           MOV     #TDEC2,4        ;TRAP POINTER
022174  005746                           TST     -(6)            ;WITH R6 = 150 SHOULD TRAP
022176  020627  000142           TDEC2:  CMP     %6,#142         ;DID R6 DECREMENT
022202  001401                           BEQ     TS313
022204  104000                           EMT                     ;R6 NOT = 142,OR WRONG $TSTNM

                                 ;*************************************************************************
                                 ;TEST 313        TEST DIFFERENT TYPES OF OVERFLOW
                                 ;*************************************************************************
022206                           TS313:
022206  010237  000146                   MOV     R2,@#146        ;RESTORE LOCATIONS 142-6
022212  010137  000144                   MOV     R1,@#144        ;
022216  010037  000142                   MOV     R0,@#142        ;
022222  012706  000150                   MOV     #150,%6
022226  005067  155714                   CLR     146             ;STATUS WORD OF LOC 10
022232  012767  022242  155544           MOV     #TDEC3,4        ;RETURN TO LOC 4
022240  005246                           INC     -(6)
022242  005767  155700           TDEC3:  TST     146
022246  001001                           BNE     1$
022250  104000                           EMT                     ;INCREMENT OPERATION NOT INHIBITED
022252  012705  001000           1$:     MOV     #1000,%5
022256  012706  000400                   MOV     #400,%6
022262  012767  022274  155514           MOV     #TDEC4,4
022270  124645                           CMPB    -(6),-(5)
022272  104000                           EMT                     ;STACK = 400 AND DECREMENTED, SHOULD TRAP
022274  012706  000400           TDEC4:  MOV     #400,%6
022300  012767  022312  155476           MOV     #TDEC7,4
022306  134546                           BITB    -(5),-(6)
022310                           TDEC6:
022310  104000                           EMT                     ;NO STACK OVERFLOW,OR WRONG $TSTNM
022312                           TDEC7:

                                 ;*************************************************************************
                                 ;TEST 314        TEST THAT AN EMT CAUSES AN OVERFLOW TRAP
                                 ;*************************************************************************
022312                           TS314:
022312  012706  000400                   MOV     #400,%6         ;SET UP STACK TO OVERFLOW
022316  012767  022334  155464           MOV     #VDEC2,10       ;SET UP 77 VECTOR
022324  012767  022340  155452           MOV     #VDEC,4 ;SET UP OVERFLOW VECTOR
022332  104000                           EMT                     ;THIS TRAP SHOULD CAUSE OVERFLOW
022334  000167  157522           VDEC2:  JMP     ERROR1          ;USE JUMP TO GET TO ERROR BECAUSE UNSURE WHAT EMT WILL D
```

```
022340  012767  021260  155442  VDEC:   MOV     #T010,10        ;RESTORE VECTOR
                                ;*******************************************************************************
                                ;TEST 315       TEST THAT AN EMT CAUSES AN OVERFLOW TRAP
                                ;*******************************************************************************
022346                          TS315:
022346  012706  000400                  MOV     #400,%6         ;SET UP STACK TO OVERFLOW
022352  012767  022370  155440          MOV     #VDEC4,20       ;SET UP IOT VECTOR
022360  012767  022374  155416          MOV     #VDEC3,4        ;SET UP OVERFLOW VECTOR
022366  104000                          EMT                     ;THIS TRAP SHOULD CAUSE OVERFLOW
022370  000167  157466          VDEC4:  JMP     ERROR1          ;USE JUMP TO GET TO ERROR BECAUSE UNSURE WHAT EMT WILL D
022374  012767  021264  155416  VDEC3:  MOV     #T020,20        ;RESTORE VECTOR

                                ;*******************************************************************************
                                ;TEST 316       TEST THAT AN EMT CAUSES AN OVERFLOW TRAP (CHECK OF YELLOW ZONE)
                                ;*******************************************************************************
022402                          TS316:
022402  012706  000400                  MOV     #400,%6         ;SET UP STACK TO OVERFLOW
022406  012767  022424  155414          MOV     #VDEC6,30       ;SET UP INST VECTOR
022414  012767  022430  155362          MOV     #VDEC5,4        ;SET UP OVERFLOW BECTOR
022422  104000                          EMT                     ;THIS TRAP SHOULD CAUSE OVERFLOW
022424  000167  157432          VDEC6:  JMP     ERROR1          ;USE JUMP TO GET TO ERROR BECAUSE UNSURE WHAT EMT WILL D
022430  012767  002062  155372  VDEC5:  MOV     #ERROR1,30      ;RESTORE VECTOR

                                ;*******************************************************************************
                                ;TEST 317       TEST THAT AN EMT CAUSES AN OVERFLOW TRAP
                                ;*******************************************************************************
022436                          TS317:
022436  012706  000400                  MOV     #400,%6         ;SET UP STACK TO OVERFLOW
022442  012767  022460  155364          MOV     #VDEC8,34       ;SET UP TRAP VECTOR
022450  012767  022464  155326          MOV     #VDEC7,4        ;SET UP OVERFLOW VECTOR
022456  104000                          EMT                     ;THIS TRAP SHOULD CAUSE OVERFLOW
022460  000167  157376          VDEC8:  JMP     ERROR1          ;USE JUMP TO GET TO ERROR BECAUSE UNSURE WHAT EMT WILL D
022464  012767  021270  155342  VDEC7:  MOV     #T034,34        ;RESTORE VECTOR
                                ;*******************************************************************************
                                ;TEST 320       TEST THAT AN EMT CAUSES AN OVERFLOW TRAP
                                ;*******************************************************************************
022472                          TS320:
022472  012706  000400                  MOV     #400,%6         ;SET UP STACK TO OVERFLOW
022476  012767  022514  155310          MOV     #VDEC10,14      ;SET UP TRT VECTOR
022504  012767  022520  155272          MOV     #VDEC9,4        ;SET UP OVERFLOW VECTOR
022512  104000                          EMT                     ;THIS TRAP SHOULD CAUSE OVERFLOW
022514  000167  157342          VDEC10: JMP     ERROR1          ;USE JUMP TO GET TO ERROR BECAUSE UNSURE WHAT EMT WILL D
022520  012767  021262  155266  VDEC9:  MOV     #T014,14        ;RESTORE VECTOR
                                ;*******************************************************************************
                                ;TEST 321       TEST THAT AN EMT CAUSES AN OVERFLOW TRAP
                                ;*******************************************************************************
022526                          TS321:
022526  012706  000400                  MOV     #400,%6         ;SET UP STACK TO OVERFLOW
022532  012767  022550  155244          MOV     #VDEC11,4       ;SET UP ILLA VECTOR
022540  012767  022554  155236          MOV     #VDEC12,4       ;SET UP OVERFLOW VECTOR
022546  104000                          EMT                     ;THIS TRAP SHOULD CAUSE OVERFLOW
022550  000167  157306          VDEC11: JMP     ERROR1          ;USE JUMP TO GET TO ERROR BECAUSE UNSURE WHAT EMT WILL D
022554  012767  021256  155222  VDEC12: MOV     #T04,4  ;RESTORE VECTOR
022562  020627  000370                  CMP     %6,#370         ;STACK PUSHED FOUR WORDS?
022566  001401                          BEQ     TS322
022570  104000                          EMT                     ;TRAP OVERFLOW DID NOT OCCUR
```

```
                                        ;*****************************************************************************
                                        ;TEST 322       TEST THAT AN EMT CAUSES AN OVERFLOW TRAP
                                        ;*****************************************************************************
022572                                  TS322:
022572  012706  000400                          MOV     #400,%6                 ;SET UP STACK TO OVERFLOW
022576  012767  022614  155200                  MOV     #VDEC13,4               ;SET UP ILLB VECTOR
022604  012767  022620  155172                  MOV     #VDEC14,4               ;SET UP OVERFLOW VECTOR
022612  104000                                  EMT                             ;THIS TRAP SHOULD CAUSE OVERFLOW
022614  000167  157242                  VDEC13: JMP     ERROR1                  ;USE JUMP TO GET TO ERROR BECAUSE UNSURE WHAT EMT WILL D
022620  012767  021256  155156          VDEC14: MOV     #T04,4  ;RESTORE VECTOR

                                        ;*****************************************************************************
                                        ;TEST 323       TEST FOR FALSE OVERFLOW TRAP
                                        ;*****************************************************************************
022626                                  TS323:

022626  012706  001000                          MOV     #STBOT,SP               ;INITIALIZE STACK
022632  012767  022700  155144                  MOV     #FOVER,4                ;SET UP OVERFLOW POINTER
022640  012706  001002                          MOV     #1002,%6
022644  005746                                  TST     -(6)                    ;SHOULD NOT OVERFLOW
022646  012706  002002                          MOV     #2002,%6
022652  005746                                  TST     -(6)                    ;SHOULD NOT OVERFLOW
022654  012706  004002                          MOV     #4002,%6
022660  005746                                  TST     -(6)                    ;SHOULD NOT OVERFLOW
022662  012706  010002                          MOV     #10002,%6
022666  005746                                  TST     -(6)
022670  012706  020000                          MOV     #20000,%6               ;SHOULD NOT OVERFLOW
022674  005746                                  TST     -(6)
022676  000401                                  BR      STP
022700                                  FOVER:
022700  104000                                  EMT                             ;IT OVERFLOWED,OR WRONG $TSTNM
022702  012767  021256  155074          STP:    MOV     #T04,4
022710  005067  155072                          CLR     6
                                        ;*****************************************************************************
                                        ;TEST 324       TEST THAT BIT 4 PSW WILL CAUSE A TRAP TO 14
                                        ;*****************************************************************************
022714                                  TS324:
022714  012706  001000                          MOV     #STBOT,SP
022720  012767  022744  155066                  MOV     #RETAT,RTRAP4           ;SET UP TO TRAP TO 14
022726  012746  000020                          MOV     #20,-(SP)               ;PUSH T BIT
022732  012746  022740                          MOV     #.+6,-(SP)              ;PUSH PC
022736  000002                                  RTI                             ;SET T BIT
022740  000240                                  NOP                             ;TRAP HERE
022742  104000                                  EMT                             ;TRACE BIT DID NOT TRAP!,OR WRONG $TESTN
022744                                  RETAT:
                                        ;*****************************************************************************
                                        ;TEST 325       TEST  STACK POINTER DECREMENTS
                                        ;*****************************************************************************
022744                                  TS325:
022744  012706  001000                          MOV     #STBOT,SP
022750  012767  022774  155036                  MOV     #RETBT,RTRAP4
022756  012746  000020                          MOV     #20,-(SP)               ;PUSH T BIT
022762  012746  022770                          MOV     #.+6,-(SP)              ;PUSH PC
022766  000002                                  RTI                             ;SET T BIT
022770  000240                                  NOP                             ;TRAP HERE
022772  104000                                  EMT                             ;TRACE BIT DID NOT TRAP!
```

```
022774  020627  000774          RETBT:  CMP     SP,#STBOT-4
023000  001401                          BEQ     TS326
023002  104000                          EMT                     ;STACK POINTER WAS NOT PUSHED BY TRAP,OR WRONG $TESTN
                                ;*****************************************************************************
                                ;TEST 326       TEST  FOR PROPER PC ON STACK
                                ;*****************************************************************************
023004                          TS326:
023004  012706  001000                  MOV     #STBOT,SP
023010  012767  023030  154776          MOV     #RETCT,RTRAP4
023016  012746  000020                  MOV     #20,-(SP)       ;PUSH T BIT
023022  012746  023030                  MOV     #.+6,-(SP)      ;PUSH PC
023026  000002                          RTI                     ;SET T BIT
                                                                ;TRAP HERE
023030  022767  023030  155736  RETCT:  CMP     #.,STBOT-4
023036  001401                          BEQ     TS327
023040  104000                          EMT                     ;CORRECT PC WAS NOT SAVED ON STACK,OR WRONG $TESTN


                                ;*****************************************************************************
                                ;TEST 327       TEST  THAT RTT POPS T- BIT
                                ;*****************************************************************************
023042                          TS327:

023042  012706  001000                  MOV     #STBOT,SP
023046  005001                          CLR     R1              ;CLEAR R1
023050  012746  000020                  MOV     #20,-(SP)
023054  012746  023070                  MOV     #RTT1,-(SP)
023060  012767  023076  154726          MOV     #RTT2,14
023066  000006                          RTT
023070  000240                  RTT1:   NOP
023072  001401                          BEQ     TS330
023074  104000                          EMT                     ;T-BIT DID NOT TRAP,OR WRONG $TESTN

023076                          RTT2:
                                ;*****************************************************************************
                                ;TEST 330       TEST  THAT RTT ALLOWS ONE INST. BEFORE TRAP
                                ;*****************************************************************************
023076                          TS330:
023076  012705  177777                  MOV     #177777,%5
023102  012706  001000          RTT5:   MOV     #STBOT,SP
023106  012746  000020                  MOV     #20,-(SP)
023112  012746  023130                  MOV     #RTT3,-(SP)
023116  012767  023140  154670          MOV     #RTT4,14
023124  005001                          CLR     R1              ;CLEAR R0
023126  000006                          RTT                     ;SET T-BIT
023130  005201                  RTT3:   INC     R1
023132  005205                          INC     %5
023134  001762                          BEQ     RTT5            ;DO THIS TEST NO MORE THAN 2 TIMES
023136  104000                          EMT                     ;DID NOT TRAP
023140  005301                  RTT4:   DEC     R1              ;SEE IF RTT ALLOWS 1 INST.
023142  001403                          BEQ     RTT6
023144  005205                          INC     %5              ;DO THIS TEST NO MORE THAN TWO TIMES
023146  001755                          BEQ     RTT5
023150  104000                          EMT                     ;RTT DID NOT ALLOW 1 INST.,OR WRONG $TESTN
023152                          RTT6:
                                ;*****************************************************************************
```

```
                                     ;TEST 331       TEST  THAT RTI DOES NOT ALLOW 1 INST.
                                     ;*****************************************************************************
023152                               TS331:
023152  012706  001000                       MOV     #STBOT,SP
023156  012746  000020                       MOV     #20,-(SP)
023162  012746  023200                       MOV     #RTI1,-(SP)
023166  012767  023204  154620               MOV     #RTI2,14
023174  005001                               CLR     R1
023176  000002                               RTI                     ;SET T-BIT
023200  005201                       RTI1:   INC     R1              ;RTI SHOULD NOT ALLOW THIS
023202  104000                               EMT                     ;T- BIT DID NOT CAUSE TRAP
023204  005701                       RTI2:   TST     R1
                                                                     ;RTI SHOULD NOT ALLOW 1 INST. BEFORE TRAP
023206  001401                               BEQ     TS332
023210  104000                               EMT                     ;RTI DID ALLOW 1 INST. BEFORE TRAP,OR WRONG $TESTN

                                     ;*****************************************************************************
                                     ;TEST 332       TEST TRAP ON TRAP THAT TRACE BIT TRAPS ARE INHIBITED ON TRAP INST
                                     ;*****************************************************************************
023212                               TS332:

023212  012706  001000                       MOV     #STBOT,%6
023216  012767  023256  154570               MOV     #TRACE,14       ;TRACE TRAP
023224  005027  000016                       CLR     #16
023230  005027  000022                       CLR     #22
023234  012767  023262  154556               MOV     #TONT1,20       ;IOT TRAP
023242  012746  000020                       MOV     #20,-(SP)       ;PUSH T BIT
023246  012746  023254                       MOV     #.+6,-(SP)      ;PUSH PC
023252  000006                               RTT
023254  000004                               IOT                     ;TRAP, NEW CC HAVE TRACE RESET
023256                               TRACE:
023256  104000                               EMT                     ;TRACE TRAP WAS NOT INHIBITED
023260                               BR70:
023260  104000                               EMT                     ;WRONG TSTNM,OR WRONG $TSTNM
023262  012767  021262  154524       TONT1:  MOV     #TO14,14        ;RESTORE BPT INSTRUCTION TRAP
023270  012767  021264  154522               MOV     #TO20,20        ;RESTORE IOT INSTRUCTION TRAP
                                     ;*****************************************************************************
                                     ;TEST 333       TEST THAT THE TRACE BIT IS SAVED IN THE STACK
                                     ;*****************************************************************************
023276                               TS333:
023276  012706  001000                       MOV     #STBOT,%6       ;SET UP STACK POINTER
023302  012767  023326  154504               MOV     #TRC1,14        ;TRACE TRAP RETURN
023310  005067  154502                       CLR     16
023314  012746  000020                       MOV     #20,-(SP)       ;SET THE T BIT
023320  012746  023326                       MOV     #TRC1,-(SP)
023324  000002                               RTI
023326  036727  155444  000020       TRC1:   BIT     STBOT-2,#20     ;CHECK FOR T BIT ON STACK
023334  001001                               BNE     STP3D
023336                               STP3:
023336  104000                               EMT                     ;T BIT NOT SAVED ON THE STACK,OR WRONG $TSTNM
023340  012767  021262  154446       STP3D:  MOV     #TO14,14

                                     ;*********************************************************************
                                     ;THIS ROUTINE TESTS THAT NO LEGAL ADDRESS TRAPS AND THAT AN ILLEGAL
                                     ;ADDRESS TRAPS TO LOCATION 4.  THIS WILL RUN ON 30K SYSTEM.  BUT IF
                                     ;SWITCH REGISTER BIT 1=0, THEN THE MEMORY FROM 28K-30K IS NOT LOOKED
```

```
                                    ;AT, SINCE IT MAY HAVE I/O DEVICES.  IF SWR BIT 1=1, THEN THAT AREA IS
                                    ;CHECKED.  (IT SHOULD EITHER ALL TRAP OR ALL NOT TRAP).  LOC 160000
                                    ;IS NO LONGER GUARANTEED TO TRAP, SINCE IT MAY CONTAIN MEMORY. LOCATION
                                    ;177700 (THE UNIBUS ADDRESS FOR RO ON OLDER SYSTEMS) IS USED FOR FORCING
                                    ;A TIMEOUT IN THE EVENT THAT THERE WAS NO TIMEOUT FROM 0K-28K OR 30K.
                                    ;THIS ROUTINE TESTS MEMORY UNTIL IT DOES A NXM STOP
                                    ;**************************************************************************
                                    ;TEST 334       TEST NON-EXISTENT ADDRESS TRAPS
                                    ;**************************************************************************
023346                              TS334:
023346  005000                      1$:     CLR     RO              ;
023350  005067  154432                      CLR     6               ;
023354  012767  023440  154422              MOV     #ATRAP,4        ;SET UP ADDRESS TRAP ENTRANCE
023362  012706  001000                      MOV     #STBOT,SP       ;SET STACK POINTER
023366  105720                      NOR:    TSTB    (0)+            ;IF OUTSIDE OF CORE, TRAP TO 4
023370  020027                              CMP     RO,(PC)+        ;IS POINTER INSIDE 28K (30K) CORE
023372  160000                      HICORE: .WORD   160000          ;MAY BE CHANGED TO 170000 IF 30K
023374  103774                              BLO     NOR             ;TEST THE REST OF COPE
023376  012737  023412  000004              MOV     #ROTRAP,@#4     ;SET UP NEW VECTOR POINTER
023404  105737  177700                      TSTB    @#177700        ;SHOULD CAUSE A TRAP
023410                              TRPADR:
023410  104000                              EMT                     ;SHOULD HAVE TRAPED
                                    ;TRAP TO HERE IF FORCING TRAP BY TESTING 177700
023412  106767  154360              ROTRAP: MFPS    STATUS
023416  105767  154354                      TSTB    STATUS          ;TEST PSW
023422  001401                              BEQ     1$
023424  104000                              EMT                     ;NEW PSW SHOULD HAVE BEEN ZERO
023426  026727  155342  023410      1$:     CMP     STBOT-4,#TRPADR ;TEST OLD PC AT STACK
023434  001437                              BEQ     TRAPB
023436  104000                              EMT                     ;OLD PC WAS NOT SAVED
                                    ;RETURN HERE ON AN ADDRESS TRAP FROM MEMORY BELOW 28K (OR 30K)
023440  005300                      ATRAP:  DEC     RO              ;
023442  010067  000032                      MOV     RO,CORH         ;MOVE THE FIRST NXM LOCATION IN CORH
                                    ;THIS ROUTINE DOES NXM TRAPS UNTIL IT FINDS AN EXISTENT MEMORY LOCATION
023446  013700  023372                      MOV     @#HICORE,RO     ;SET UP THE HIGHEST MEM LOCATION
023452  005300                              DEC     RO              ;MAKE 1 LESS THAN THE HIGHEST CORE BOUNDARY
023454  000402                              BR      NOSUB           ;DON'T SUBTRACT 1K FIRST TIME
023456  162700  001000              CTRAP:  SUB     #1000,RO        ;SUBTRACT 1K OCTAL BYTE FROM ADDRESS
                                                                    ;TO SPEED UP TESTING
023462  012767  023506  154314      NOSUB:  MOV     #BTRAP,4        ;SET UP THE VECTOR
023470  012706  001000                      MOV     #STBOT,SP
023474  005710                              TST     (RO)            ;DOES THIS MEMORY EXIST?
                                                                    ;IF NXM, TRAP TO BTRAP
023476  020027                      DTRAP1: CMP     RO,(PC)+        ;IF EXISTS, IS THIS THE SAME TRAP THAT CAUSED
                                                                    ;TRAP TO ATRAP
023500  000000                      CORH:   .WORD   0
023502  101414                              BLOS    TRAPB
023504  104000                              EMT                     ;CONTENTS OF RO SHOULD BE LESS THAN OR EQUAL  TO CORH
                                                                    ;IF THIS COMPARISON FAILS IT MEANS
                                                                    ;THAT SOME LEGAL ADDRESS TRAPPED, OR
                                                                    ;THAT AN ILLEGAL ADDRESS DID NOT TRAP
023506  106767  154264              BTRAP:  MFPS    STATUS
023512  105767  154260                      TSTB    STATUS
023516  001401                              BEQ     1$
023520  104000                              EMT                     ;NEW PSW SHOULD HAVE BEEN ZERO
023522  026727  155246  023476      1$:     CMP     STBOT-4,#DTRAP1 ;CHECK IF TRAP PC IS OK
```

```
023530  001752                                  BEQ     CTRAP
023532                                  AUTO1:
023532  104000                                  EMT                     ;OLD PC WAS NOT SAVED OR WRONG $TESTN
023534  012767  021256  154242          TRAPB:  MOV     #T04,4          ;RESET TRAP CATCHER
023542  005067  154240                          CLR     6               ;RESET TRAP CATCHER


                                        ;THIS ROUTINE WILL FIGURE OUT IF YOU HAVE A DL11W
023546  012706  001000                          MOV     #STBOT,SP       ;SET UP THE STACK POINTER
023552  012767  023566  154224                  MOV     #NODL,4         ;SET UP THE TRAP VECTOR
023560  005767  154000                          TST     TTCSR           ;TEST THE PUNCH STATUS REGISTER
023564  000405                                  BR      DL11W
023566  012767  021256  154210          NODL:   MOV     #T04,4
023574  000167  074760                          JMP     SLU1ST          ;IF NO SLU FIND OUT WHY IN SLU TEST
023600  012767  021256  154176          DL11W:  MOV     #T04,4

                                        ;*************************************************************************
                                        ;TEST 335       TEST THAT A TTY INTERRUPT CAUSES AN OVERFLOW TRAP
                                        ;*************************************************************************
023606                                  TS335:
023606  032737  000001  001020                  BIT     #1,@#$ENV       ;ARE WE RUNNING UNDER APT?
023614  001405                                  BEQ     1$              ;IF NO THEN GO DO THE TEST
023616  005737  001006                          TST     @#$PASS         ;IS THIS THE FIRST PASS?
023622  001402                                  BEQ     1$              ;YES,GO DO THE TEST
023624  000167  000406                          JMP     TS341           ;NO SKIP THE NEXT FOUR TESTS
                                                                        ;BECAUSE APT USES SLU1 TO COMMUNICATE
023630  012767  000340  154140          1$:     MOV     #340,STATUS     ;LOCK OUT INTERRUPT
023636  012706  000400                          MOV     #400,%6         ;SET UP STACK TO OVERFLOW
023642  012767  023704  154134                  MOV     #TDEC77,4       ;SET UP OVERFLOW TRAP
023650  016767  154210  001542                  MOV     64,     TEMP1   ;SAVE CONTENTS OF INTERRUPT VECTOR
023656  012767  023702  154200                  MOV     #TDEC8,64       ;SET UP INTERRUPT VECTOR
023664  012767  000100  153672                  MOV     #100,TTCSR      ;SET INTERRUPT ENABLE
023672  005067  154100                          CLR     STATUS          ;ALLOW INTERRUPT TO OCCUR
023676  000167  074656                          JMP     SLU1ST          ;NO INTERRUPT OCCURRED SO GO TO SLU TEST
                                                                        ;TO FIND OUT WHY ADD REPORT PROPER ERROR
023702                                  TDEC8:
023702  104000                                  EMT                     ;OVERFLOW TRAP DID NOT OCCUR
023704  005067  153654                  TDEC77: CLR     TTCSR           ;CLEAR INTERRUPT ENABLE
023710  012767  021256  154066                  MOV     #T04,4
023716  005067  154064                          CLR     6
023722  016767  001472  154134                  MOV     TEMP1,  64      ;RESTORE CONTENTS OF INTERRUPT VECTOR
                                        ;*************************************************************************
                                        ;TEST 336       TEST THAT A PENDING INTERRUPT OCCURS BEFORE TRAP
                                        ;*************************************************************************
023730                                  TS336:
023730  012706  001000                          MOV     #STBOT,%6
023734  012767  000340  154034                  MOV     #340,STATUS     ;SET TO A HIGH PRIORITY LEVEL
023742  016767  154116  001450                  MOV     64,     TEMP1   ;SAVE CONTENTS OF INTERRUPT VECTOR
023750  012767  024014  154106                  MOV     #TR0,64
023756  012767  000100  153600                  MOV     #100,TTCSR      ;INTERRUPT FOR TTY PUNCH/PRINTER
023764  012767  024016  154042                  MOV     #BR71,34        ;TRAP VECTOR
023772  012767  024020  154064                  MOV     #TR2,64         ;TTY VECTOR
024000  012767  000340  154030                  MOV     #340,36         ;IF TRAP TRAPS, MOVE 340 TO PRIORITY
024006  005067  153764                          CLR     STATUS          ;SHOULD INTERRUPT AT END OF CLR INST
024012  104400                                  TRAP                    ;TTY INTERRUPT SHOULD OVERRIDE TRAP
024014                                  TR0:
```

```
024014  104000                                   EMT                        ;TTY SHOULDN'T HAVE INTERRUPTED
024016                                   BR71:
024016  104000                                   EMT                        ;TRAP OCCURRED FIRST
024020  005067  154012                   TR2:    CLR     36
024024  016767  001370  154032                   MOV     TEMP1,  64         ;RESTORE CONTENTS OF INTERRUPT VECTOR
024032  042767  000100  153524                   BIC     #100,TTCSR
                                         ;*******************************************************************************
                                         ;TEST 337       TEST THAT A PENDING INTERRUPT, INTERRUPTS BETWEEN TRAPS
                                         ;*******************************************************************************
024040                                   TS337:
024040  012706  001000                           MOV     #STBOT,%6
024044  012767  000340  153724                   MOV     #340,STATUS
024052  012767  000100  153504                   MOV     #100,TTCSR
024060  012767  024126  153746                   MOV     #TR3,34            ;TRAP
024066  016767  153772  001324                   MOV     64,     TEMP1      ;SAVE CONTENTS OF INTERRUPT VECTOR
024074  012767  024132  153762                   MOV     #TR4,64            ;TTY OUTPUT
024102  012767  000340  153756                   MOV     #340,66            ;TTY OUTPUT PRIORITY
024110  012767  024130  153702                   MOV     #TR5,20            ;IOT
024116  012767  000340  153676                   MOV     #340,22            ;IOT PRIORITY
024124  104400                                   TRAP                       ;THE ACT OF TRAPPING LOWER PRIORITY
024126  000004                           TR3:    IOT                        ;INTERRUPT SHOULD OCCUR IN PLACE OF IOT TRAP
024130                                   TR5:
024130  104000                                   EMT                        ;NO INTERRUPT BETWEEN TRAPS,OR WRONG $TSTNM
024132  005067  153664                   TR4:    CLR     22                 ;CLR IOT PRIORITY
024136  005067  153724                           CLR     66
024142  012767  021270  153664                   MOV     #TO34,34
024150  016767  001244  153706                   MOV     TEMP1,  64         ;RESTORE CONTENTS OF INTERRUPT VECTOR
024156  012767  000022  153634                   MOV     #22,20
024164  042767  000100  153372                   BIC     #100,TTCSR         ;CLEAR IE BIT IN SLU1 XMIT CSR

                                         ;*******************************************************************************
                                         ;TEST 340       TEST THAT "RESET" GOES TO OUTSIDE WORLD
                                         ;*******************************************************************************
024172                                   TS340:
024172  032737  000001  001020                   BIT     #1, @#$ENV         ;ARE WE RUNNING UNDER APT
024200  001403                                   BEQ     70$                ;IF NO THEN DO TEST
024202  005737  001006                           TST     @#$PASS            ;IS THIS FIRST PASS
024206  001013                                   BNE     TS341              ;IF NO THEN SHIP TO NEXT TEST
024210                                   70$:
024210  016700  153346                           MOV     TKB,R0             ;MAKE SURE RECEIVER DONE IS CLEAR
024214  012767  000100  153336                   MOV     #100,TRCSR         ;SET INTERRUPT ENABLE
024222  000005                                   RESET                      ;SHOULD CLEAR INTERRUPT ENABLE
024224  032767  000100  153326                   BIT     #100,TRCSR         ;TEST FOR CLEAR
024232  001401                                   BEQ     TS341
024234  104000                                   EMT                        ;RESET FAILED TO CLEAR TRCSR,OR WRONG $TSTNM
024236                                   NODL2:
                                         ;*******************************************************************************
                                         ;TEST 341       TEST THAT RESET HAS NO EFFECT ON THE TRACE TRAP
                                         ;*******************************************************************************
024236                                   TS341:
024236  032737  000001  001020                   BIT     #1, @#$ENV         ;ARE WE RUNNING UNDER APT
024244  001403                                   BEQ     70$                ;IF NO THEN DO TEST
024246  005737  001006                           TST     @#$PASS            ;IS THIS FIRST PASS
024252  001024                                   BNE     TS342              ;IF NO THEN SHIP TO NEXT TEST
024254                                   70$:
024254  012706  001000                           MOV     #STBOT,%6          ;SET STACK
```

```
024260  012767  024306  153526            MOV     #RESET2,14        ;SET UP TRACE VECTOR
024266  012746  000020                    MOV     #20,-(R6)         ;SET THE T-BIT ON STACK
024272  012746  024300                    MOV     #1$,-(R6)         ;MOVE NEW PC ON STACK
024276  000006                            RTT
024300  000005                    1$:     RESET                     ;SHOULD HAVE NO EFFECT
024302  000005                            RESET                     ;NO EFFECT
024304                            RESET3:
024304  104000                            EMT                       ;TRACE TRAP FAILED,OR WRONG $TSTNM
024306  005067  153464            RESET2: CLR     STATUS            ;CLEAR TRACK
024312  005067  153500                    CLR     16                ;TRACE STATUS
024316  012767  021262  153470            MOV     #T014,14
024324                            SKTST2:

                                  ;*****************************************************************************
                                  ;TEST 342       TEST THE 'WAIT' INSTRUCTION
                                  ;*****************************************************************************
024324                            TS342:
024324  122767  000001  154466            CMPB    #APTENV,$ENV              ;RUNING IN APT MODE?
024332  001003                            BNE     1$                        ;IF NOT, DO THIS TEST
024334  005767  154446                    TST     $PASS                     ;IS THIS THE FIRST PASS?
024340  001051                            BNE     STP4E                     ;IF NOT FIRST PASS, SKIP TEST
024342                            1$:
024342  042767  000100  153214            BIC     #100,TTCSR        ;CLEAR INTERRUPT ENABLE
024350  012706  001000                    MOV     #STBOT,SP         ;SET UP THE STACK
024354  016767  153504  001036            MOV     64,     TEMP1     ;SAVE CONTENTS OF INTERRUPT VECTOR
024362  012767  024442  153474            MOV     #WATE,64          ;SET UP THE INTERRUPT VECTOR
024370  005067  153472                    CLR     66
024374  105767  153164            WATE1:  TSTB    TTCSR             ;WAIT FOR READY
024400  100375                            BPL     WATE1             ;TO BE UP
024402  012767  000015  153156            MOV     #15,TPB           ;DO A CARRIAGE RETURN
024410  105767  153150            WATE2:  TSTB    TTCSR             ;WAIT FOR READY TO COME UP
024414  100375                            BPL     WATE2
024416  012767  000015  153142            MOV     #15,TPB           ;DO ANOTHER CARRIAGE RETURN
024424  052767  000100  153132            BIS     #100,TTCSR        ;SET THE INTERRUPT ENABLE
024432  005067  153340                    CLR     STATUS            ;CLEAR THE PSW
024436  000001                    WATE3:  WAIT                      ;WAIT FOR THE INTERRUPT
024440  104000                            EMT                       ;WAIT INSTRUCTION DID NOT LOOP
024442  005767  153330            WATE:   TST     STATUS  ;IS THE PSW CORRECT?
024446  001401                            BEQ     1$
024450  104000                            EMT                       ;NEW PSW SHOULD HAVE BEEN ZERO
024452  026727  154316  024440    1$:     CMP     STBOT-4,#WATE3+2          ;IS THE OLD PC SAVED
024460  001401                            BEQ     STP4E
024462                            STP4:
024462  104000                            EMT                       ;OLD PC WAS NOT SAVED OR WRONG $TESTN
024464  016767  000730  153372    STP4E:  MOV     TEMP1,  64        ;RESTORE CONTENTS OF INTERRUPT VECTOR
024472  042767  000100  153064            BIC     #100,TTCSR        ;CLEAR IE BIT IN SLU1 XMIT CSR
                                  ;*****************************************************************************
                                  ;TEST 343       TEST THAT USING REGISTER ADDR (177700) CAUSES TIME OUT.
                                  ;*****************************************************************************
024500                            TS343:
                                          ;
                                          ;REGISTER ADDRESS (177700-177717) CAUSE TIME OUT WHEN USED
                                          ;AS PROGRAM ADDRESS BY THE CPU.
                                          ;
024500  012706  001000                    MOV      #STBOT,SP        ;SET  STACK POINTER
024504  012767  024520  153272            MOV      #RETR1,RTRAP5    ;SET TRAP RETURN ADDR
```

```
024512  005237  177700          PCN1:   INC     @#177700        ;BAD ADDR REFERENCE, TRAP TO 4
024516  104000                          EMT                     ;REFERENCING 177700 DID NOT CAUSE TIME OUT
024520  022767  024516  154246  RETR1:  CMP     #PCN1+4,STBOT-4 ;PROPER PC STORED ON STACK?
024526  001401                          BEQ     TS344
024530  104000                          EMT                     ;OLD PC WAS NOT SAVED IN STACK

                                        ;*******************************************************************
                                        ;ODD ADDRESS USED BY A "WORD" INSTRUCTION SHOULD NOT
                                        ;CAUSE A TRAP, BUT THE LOW ORDER ADDRESS BIT WOULD BE IGNORED.
                                ;***************************************************************************
                                ;TEST 344       TEST ODD ADDRESS TRAP IS NOT IMPLEMENTED.
                                ;***************************************************************************
024532                          TS344:
                                        ;
024532  012737  024556  000004          MOV     #RETR2,@#RTRAP5 ;SET TRAP RETURN ADDR
024540  005037  000000                  CLR     @#0             ;PUT ALL 0 IN LOC 0
024544  005337  000001                  DEC     @#1             ;DECREMENT ODD ADDRESS, SHOULD NOT TRAP
024550  022737  177777  000000          CMP     #-1,@#0         ;WORD LOC 0 HAS ALL ONES?
024556                          RETR2:
024556  001401                          BEQ     TS345
024560  104000                          EMT                     ;LOC 0 DID NOT STORE -1,OR ODD ADDR REFERENCE CAUSE TRAP

                                        ;*******************************************************************
                                        ;
                                        ;USING ADDRESS 177700 IN MODE 2, CAUSES BUS ERROR, BUT
                                        ;THE REGISTER IN USE WILL BE INCREMENTED.
                                        ;
                                ;***************************************************************************
                                ;TEST 345       TEST THAT IN MODE 2, BAD ADDRESS REFERENCE CAUSES BUS ERROR.
                                ;***************************************************************************
024562                          TS345:
024562  012737  024602  000004          MOV     #RETR3,@#RTRAP5 ;SET TRAP RETURN ADDR
024570  012700  177700                  MOV     #177700,R0      ;STORES BAD MEMORY REFERENCE
024574  012720  001234                  MOV     #1234,(R0)+     ;BAD ADDR REFERENCE, TRAP TO LOC 4
024600  104000                          EMT                     ;ADDRESSING 177700 DID NOT CAUSE TRAP
024602  022700  177702          RETR3:  CMP     #177702,R0      ;WAS R0 INCREMENTED?
024606  001401                          BEQ     TS346
024610  104000                          EMT                     ;R0 WAS NOT INCREMENTED
                                        ;*******************************************************************
                                        ;
                                        ;AFTER THE FIRST BUS ERROR WAS ENCOUNTERED, AN ATTEMPT WAS MADE
                                        ;TO PUSH PC AND PS INTO THE STACK. HOWEVER, IF THE STACK POINTER
                                        ;WAS BAD, A BOUBLE BUS ERROR OCCURED.  THE STACK POINTER WOULD
                                        ;THEN BE SET TO LOCATION 4, OLD PC AND PS WERE PUSHED INTO
                                        ;LOCATIONS 0 AND 2. THE PROCESSOR WOULD TRAP TO 4 AND CONTINUE
                                        ;EXECUTION.
                                        ;
                                ;***************************************************************************
                                ;TEST 346       TEST FOR DOUBLE BUS ERROR.
                                ;***************************************************************************
024612                          TS346:
024612  012767  024654  153164          MOV     #DBE1,RTRAP5    ;SET TRAP RETURN ADDR
024620  012737  000340  000006          MOV     #340,@#6        ;SET UP PS
024626  012767  024652  153154          MOV     #DBE2,RTRAP     ;SET TRAP RETURN ADDR
024634  012737  000340  000012          MOV     #340,@#12       ;SET UP PS
024642  012706  177700                  MOV     #177700,SP      ;SET ILLEGAL SP
```

```
024646  000077                          DBE:    TRAPA                           ;ILLEGAL INSTRUCTION
024650  104000                                  EMT                             ;DOUBLE BUS ERROR DID NOT CAUSE TRAP
024652                                  DBE2:
024652  104000                                  EMT                             ;TRAP TO WRONG LOCATION
024654  022737  024650  000000          DBE1:   CMP     #DBE+2,@#0              ;OLD PC GOT SAVED?
024662  001401                                  BEQ     DBE3
024664  104000                                  EMT                             ;OLD PC DID NOT GET SAVEDD
024666  022737  000340  000002          DBE3:   CMP     #340,@#2                ;CORRECT PS SAVED?
024674  001401                                  BEQ     DBE4
024676  104000                                  EMT                             ;CORRECT PS DID NOT GET SAVE
024700  022706  000000                  DBE4:   CMP     #0,SP                   ;SP POINTS TO LOC 0?
024704  001401                                  BEQ     DBE5
024706  104000                                  EMT                             ;SP IS NOT POINTING TO LOC 0
024710  012706  001000                  DBE5:   MOV     #STBOT,SP               ;RESET SP
024714  012767  021256  153062                  MOV     #T04,4                  ;RESET VECTOR 4
024722  012767  021260  153060                  MOV     #T010,10                ;RESET VECTOR 10

                                                ;*************************************************************
                                                ;THIS TEST WILL CHECK THE SERVICE ROUTINE FOR A CONTROL CHIP ERROR.
                                                ;THIS IS DONE BY EXECUTING INSTRUCTIONS WHICH JUMP TO NON-EXISTENT
                                                ;CONTROL-CHIP. THE TEST EXECUTES AN FIS INSTRUCTION WHICH
                                                ;IS ILLEGAL ON ALL PROCESSORS USING THE DCF11-A CHIP SET.
                                                ;A CTLERR TRAPS TO LOCATION 10.
                                                ;THE RESET LINE IS ALSO ASSERTED FOR 1 CYCLE.
                                        ;*****************************************************************************
                                        ;TEST 347       TEST CTLERR SERVICE ROUTINE
                                        ;*****************************************************************************
024730                                  TS347:

024730  012706  001000                          MOV     #STBOT,R6               ;INIT STACK POINTER
024734  012737  024754  000010                  MOV     #1$,@#10                        ;SET UP RETURN ADDR FROM TRAP
024742  012737  000340  000012                  MOV     #340,@#12               ;SET TRAP PRIORITY=7
024750  075006                                  FADD    R6                      ;EXECUTE FIS INSTR..SHOULD CAUSE CTLERR
024752  000000                                  HALT                            ;DID NOT TRAP..CHECK CSEL LINE
024754  012706  001000                  1$:     MOV     #STBOT,R6               ;RE-INIT STACK POINTER
024760  012767  021260  153022                  MOV     #T010,10                ;RESET VECTOR 10

                                        ;*****************************************************************************
                                        ;TEST 350       TEST THAT ALL RESERVED INSTRUCTIONS TRAP
                                        ;*****************************************************************************
024766                                  TS350:
024766  042767  000100  152570                  BIC     #100,TTCSR
                                                                                ; SET UP TO SEE IF
024774  013767  000010  000042                  MOV     @#10,TENSAVE            ;  THIS PROCESSOR HAS THE
025002  012737  025046  000010                  MOV     #TRAP10,@#10            ;  FLOATING POINT OPTION
025010  170127  000000                          LDFPS   #0                      ;DO A FPP INSTRUCTION
                                                                                ;IF NO TRAP FPP INSTALLED
025014  013767  025370  000356                  MOV     @#FPP,FINISH            ;SO RESET END OF TABLE POINTER
025022  000411                                  BR      AROUND                  ; THE FOLLOWING

                                        ;* IF NO CIS OPTION TRAP TO HERE
025024  042777  000040  174272          CISTRP: BIC     #40,@SWR                ;CLEAR CIS OPTION IN SWR
025032  012716  025100                          MOV     #CONCIS,(SP)            ;CHANGE RETURN ADDRESS TO CONCIS LOCATION
025036  000002                                  RTI                             ;RETURN
025040  000000                          CISADR: .WORD   0                       ;DATA FOR CIS
025042  000000                                  .WORD   0                       ;               INSTRUCTION
```

```
025044  000000                    TENSAVE:.WORD   0                 ; A PLACE TO STORE CONTENTS OF 10
025046                            TRAP10:                           ; LEAVE THE TABLE ALONE

025046                            AROUND:                           ;  CONTINUATION POINT
025046  012737  000246  000244            MOV     #246,@#244        ;  RESTORE THE TRAP VECTOR
025054  012737  025024  000010            MOV     #CISTRP,@#10      ;SET UP TO SEE IF THIS HAS THE CIS OPTION
025062  076144                            .WORD   076144            ;EXECUTE A CMPCI INSTRUCTION
025064  025040                            .WORD   CISADR            ;OPERANDS
025066  025040                            .WORD   CISADR            ;         FOR CIS
025070  000000                            .WORD   0                 ;                 INSTRUCTION
025072  052777  000040  174224            BIS     #40,@SWR          ;SET CIS PRESENT BIT
025100  016737  177740  000010    CONCIS: MOV     TENSAVE,@#10      ; RESTORE THE ILLEGAL INST. VECTOR
025106  012703  025260                    MOV     #TABLE,TAB        ;TABLE POINTER
025112  012305                    GIN1:   MOV     (TAB)+,FIRST      ;FIRST OR CURRENT INSTRUCTION
025114  012301                            MOV     (TAB)+,LAST       ;LAST INSTRUCTION OR GROUP
025116  020537  025334                    CMP     FIRST,@#CIS
025122  001007                            BNE     1$
025124  032777  000040  174172            BIT     #40,@SWR
025132  001403                            BEQ     1$
025134  012703  025370                    MOV     #FPP,TAB
025140  000764                            BR      GIN1
025142  020567  000232            1$:     CMP     FIRST,FINISH      ;TESTED ALL
025146  001415                            BEQ     GIN3              ;YES BRANCH
025150  010567  000226                    MOV     FIRST,INST        ;SET UP INST
025154  005267  000222            GIN2:   INC     INST
025160  012767  025214  152622            MOV     #RET,10           ;SET UP RETURN FROM TRAP
025166  012706  001000                    MOV     #STBOT,SP         ;SET UP STACK POINTER
025172  005067  152600                    CLR     CC                ;CLEAR PRIORITY
025176  000167  000200                    JMP     INST              ;EXECUTE RESERVED INSTRUCTION
025202  012767  021260  152600    GIN3:   MOV     #TO10,10          ;RESET VECTOR 10
025210  000167  000252                    JMP     THRPRT            ;JUMP TO EIS TEST

                                  ;TRAPPING SHOULD SEND YOU HERE
025214  020627  000774            RET:    CMP     SP,#STBOT-4       ;TEST DECREMENT OF SP
025220  001401                            BEQ     RET1
025222  104000                            EMT                       ;WRONG DECREMENT
025224  026727  153544  025404    RET1:   CMP     STBOT-4,#INST+2   ;LOC OF INST UNINCREMENTED
025232  001401                            BEQ     RET2
025234  104000                            EMT                       ;INST INC ON TRAP
025236  005767  153534            RET2:   TST     STBOT-2
025242  001401                            BEQ     RET3
025244                            RET4:
025244  104000                            EMT                       ;CONDITION CODES SET ON TRAP OR WRONG $TSTNM
025246  026701  000130            RET3:   CMP     INST,LAST
025252  001717                            BEQ     GIN1              ;SET UP NEW GROUP
025254  000167  177674                    JMP     GIN2              ;FINISH OLD GROUP
                                                                    ;END OF INSTRUCTION GROUP
025260  000007                    TABLE:  7                         ;END OF OPERATE
025262  000077                            77
025264  000207                            207                       ;RTS,RT1,JMP
025266  000227                            227
025270  006777                            6777
025272  007777                            7777
025274  075037                            075037
025276  076017                            76017
025300  076032                            76032
```

```
025302  076037                    76037
025304  076045                    76045
025306  076047                    76047
025310  076077                    76077
025312  076127                    76127
025314  076132                    76132
025316  076137                    76137
025320  076145                    76145
025322  076147                    76147
025324  076157                    76157
025326  076167                    76167
025330  076177                    76177
025332  076777                    76777
025334  076017          CIS:      76017
025336  076032                    76032
025340  076037                    76037
025342  076045                    76045
025344  076047                    76047
025346  076077                    76077
025350  076127                    76127
025352  076132                    76132
025354  076137                    76137
025356  076145                    76145
025360  076147                    76147
025362  076157                    76157
025364  076167                    76167
025366  076177                    76177
025370  167777          FPP:      167777                  ;  START OF THE FPP INSTRUCTIONS
025372  177700                    177700
025374  177716                    177716
025376  177777                    177777
025400  025400          FINISH:   .                       ;END FLAG
025402  000000          INST:     HALT                    ;WILL CONTINUE RESERVED INST
025404  000000                    HALT                    ;SHOULD TRAP TO LOC 10
025406  000000                    HALT                    ;LOC 10 SHOULD SEND YOU TO
025410  000000                    HALT                    ;RET
025412  000000                    HALT
                        .SBTTL ** STARTING OF EIS TEST **


        000000          DUMMY=    0
        000051          F=        51
        000176          N=        176

025414                  COUNT:
        025416                    .=COUNT+2
025416                  PSWORD:
        025420                    .=PSWORD+2
025420                  TEMP1:
        025422                    .=TEMP1+2
025422                  TEMP2:
        025424                    .=TEMP2+2
025424                  TEMP3:
        025426                    .=TEMP3+2
025426                  TEMP4:
        025430                    .=TEMP4+2
```

```
025430 000000                      TEMP5:  .WORD
025432 000000                      TEMP6:  .WORD
025434 177771                      S1:     -7
025436 025434                      S2:     S1
025440 177772                      S3:     -6
025442 177777                      S4:     -1
025444 040000                      S5:     40000
025446 025444                      S6:     S5
025450 040000                      S7:     40000
025452 177776                      S8:     -2
025454 000002                      S9:     2
025456 025454                      S10:    S9
025460 000002                      S11:    2
025462 177566                      $TPB:   177566
025464 177564                      $TPS:   177564







025466                             THRPRT:

025466 012705 001004                       MOV     #$TESTN,R5      ;MAKE R5 POINT TO WHERE TEST # IS SAVED
025472 005037 025414                       CLR     @#COUNT         ;CLEAR THE COUNTER
025476 012715 000001                       MOV     #1,(R5)         ;INITIALIZE TEST NUMBER
025502 012706 001000                       MOV     #STBOT,SP       ;** STACK AT STBOT **
025506 012737 000001 025420        2$:     MOV     #1,@#TEMP1      ;TEMP1=1
025514 005037 025422                       CLR     @#TEMP2         ;TEMP2=0
02552C 012737 000001 025424                MOV     #1,@#TEMP3      ;TEMP3=1
025526 005037 025426                       CLR     @#TEMP4         ;TEMP4=0
025532 106427 000000                       MTPS    #0
```

```



                                                  ;********************************
                                                  ;      ASH INSTRUCTION TESTS
                                                  ;********************************




                                                          ;*************
                                                          ; TESTS 1-36
                                                          ;*************


025536  013700  025420          ASTART: MOV     @#TEMP1,%0      ;LOAD R0 WITH THE CONTENTS OF TEMP1
025542  032737  000001  001006          BIT     #1,@#$PASS      ;IS IT AN EVEN PASS ?
025550  001004                          BNE     2$              ;IF NOT THEN GO TO 2$
025552  013701  025422                  MOV     @#TEMP2,R1      ;OTHERWISE EXECUTE THE INSTRUCTION
                                                                ;IN MODE 0 USING R1
025556  072001                          ASH     R1,R0
025560  000402                          BR      4$
025562  072067  177634          2$:     ASH     TEMP2,%0        ;SHIFT R0 BY THE NUMBER SPECIFIED BY TEMP2
025566  106737  025416          4$:     MFPS    @#PSWORD        ;SAVE PS
025572  123737  025426  025416          CMPB    @#TEMP4,@#PSWORD;IS THE PS = TEMP4 ?
025600  001401                          BEQ     11$
025602  104000                          EMT                     ;THE PS IS NOT EQUAL TO 0
025604  005237  025414          11$:    INC     @#COUNT         ;INCREMENT THE COUNTER
025610  023700  025424                  CMP     @#TEMP3,%0      ;IS THE RESULT IN R0 EQUAL TO TEMP3?
025614  001401                          BEQ     12$
025616                          6$:
025616  104000                          EMT                     ;EITHER INCORRECT R0 OR INCORRECT SEQUENCE
025620  021537  025414          12$:    CMP     (R5),@#COUNT    ;IS THE TEST NUMBER EQUAL TO THE
                                                                ;COUNTER?
025624  001374                          BNE     6$              ;IF NOT GO TO THE HLT ABOVE
025626  005215                          INC     (R5)
025630  021527  000037                  CMP     (R5),#37        ;HAS THE CONTENTS OF REGISTERS BEEN SHIFTED LEFT
                                                                ;BY 14. AND RIGHT BY 14.?
025634  002011                          BGE     8$
025636  005237  025422                  INC     @#TEMP2
025642  006367  177556                  ASL     TEMP3           ;SHIFT TEMP3 LEFT.
025646  021527  000020                  CMP     (R5),#20        ;HAS THE CONTENTS OF REGISTERS BEEN SHIFTED LEFT BY 14.?
025652  001004                          BNE     REGR1
025654  000167  000670                  JMP     NEGAT           ;IF SO GO TO NEGAT AND INITIATE RIGHT SHIFT
025660  004767  000712          8$:     JSR     PC,TST37        ;IF SO GO AND CONTINUE THE REST OF THE PROGRAM
025664  013701  025420          REGR1:  MOV     @#TEMP1,%1      ;LOAD R1 WITH THE CONTENTS OF TEMP1
025670  032737  000001  001006          BIT     #1,@#$PASS      ;IS IT AN EVEN PASS ?
025676  001004                          BNE     2$              ;IF NOT THEN GO TO 2$
025700  013702  025422                  MOV     @#TEMP2,R2      ;OTHERWISE EXECUTE ASH INSTRUCTION IN MODE 0
025704  072102                          ASH     R2,R1           ;USING R1
025706  000402                          BR      4$
025710  072167  177506          2$:     ASH     TEMP2,%1        ;SHIFT R1 BY THE NUMBER SPECIFIED BY TEMP2
025714  106737  025416          4$:     MFPS    @#PSWORD        ;SAVE PS
025720  123737  025426  025416          CMPB    @#TEMP4,@#PSWORD;IS THE PS = TEMP4 ?
025726  001401                          BEQ     11$
025730  104000                          EMT                     ;THE PS IS NOT EQUAL TO 0
025732  005237  025414          11$:    INC     @#COUNT         ;INCREMENT THE COUNTER
025736  023701  025424                  CMP     @#TEMP3,%1      ;IS THE RESULT IN R1 EQUAL TO TEMP3?
```

```
025742  001401                          BEQ     12$
025744                          6$:
025744  104000                          EMT                             ;EITHER INCORRECT R1 OR INCORRECT SEQUENCE
025746  021537  025414          12$:    CMP     (R5),@#COUNT            ;IS THE TEST NUMBER EQUAL TO THE COUNTER?
025752  001374                          BNE     6$                      ;IF NOT GO TO THE HLT ABOVE
025754  005215                          INC     (R5)
025756  021527  000037                  CMP     (R5),#37                ;HAS THE CONTENTS OF REGISTERS BEEN SHIFTED LEFT
                                                                        ;BY 14. AND RIGHT BY 14.?
025762  002011                          BGE     8$
025764  005237  025422                  INC     @#TEMP2
025770  006367  177430                  ASL     TEMP3                   ;SHIFT TEMP3 LEFT
025774  021527  000020                  CMP     (R5),#20                ;HAS THE CONTENTS OF REGISTERS BEEN SHIFTED LEFT BY 14.?
026000  001004                          BNE     REGR2
026002  000167  000542                  JMP     NEGAT                   ;IF SO GO TO NEGAT AND INITIATE RIGHT SHIFT
026006  004767  000564          8$:     JSR     PC,TST37                ;IF SO GO AND CONTINUE THE REST OF THE PROGRAM
026012  013702  025420          REGR2:  MOV     @#TEMP1,%2              ;LOAD R2 WITH THE CONTENTS OF TEMP1
026016  032737  000001  001006          BIT     #1,@#$PASS              ;IS IT AN EVEN PASS ?
026024  001004                          BNE     2$                      ;IF NOT THEN GO TO 2$
026026  013703  025422                  MOV     @#TEMP2,R3              ;OTHERWISE EXECUTE ASH INSTRUCTION IN MODE 0
026032  072203                          ASH     R3,R2                   ;USING R2
026034  000402                          BR      4$
026036  072267  177360          2$:     ASH     TEMP2,%2                ;SHIFT R2 BY THE NUMBER SPECIFIED BY TEMP2
026042  106737  025416          4$:     MFPS    @#PSWORD                ;SAVE PS
026046  123737  025426  025416          CMPB    @#TEMP4,@#PSWORD;IS THE PS = TEMP4 ?
026054  001401                          BEQ     11$
026056  104000                          EMT                             ;THE PS IS NOT EQUAL TO 0
026060  005237  025414          11$:    INC     @#COUNT
026064  023702  025424                  CMP     @#TEMP3,%2              ;IS THE RESULT IN R2 EQUAL TO TEMP3?
026070  001401                          BEQ     12$
026072                          6$:
026072  104000                          EMT                             ;EITHER INCORRECT R2 OR INCORRECT SEQUENCE
026074  021537  025414          12$:    CMP     (R5),@#COUNT            ;IS THE TEST NUMBER EQUAL TO THE COUNTER?
026100  001374                          BNE     6$                      ;IF NOT GO TO THE HLT ABOVE
026102  005215                          INC     (R5)
026104  021527  000037                  CMP     (R5),#37                ;HAS THE CONTENTS OF REGISTERS BEEN SHIFTED
                                                                        ;LEFT BY 14, AND RIGHT BY 14.?
026110  002011                          BGE     8$
026112  005237  025422                  INC     @#TEMP2
026116  006367  177302                  ASL     TEMP3                   ;SHIFTED TEMP3 LEFT
026122  021527  000020                  CMP     (R5),#20                ;HAS THE CONTENTS OF REGISTERS BEEN SHIFTED LEFT BY 14.?
026126  001004                          BNE     REGR3
026130  000167  000414                  JMP     NEGAT                   ;IF SO GO TO NEGAT AND INITIATE RIGHT SHIFT
026134  004767  000436          8$:     JSR     PC,TST37                ;IF SO GO AND CONTINUE THE REST OF THE PROGRAM
026140  013703  025420          REGR3:  MOV     @#TEMP1,%3              ;LOAD R3 WITH THE CONTENTS OF TEMP1
026144  032737  000001  001006          BIT     #1,@#$PASS              ;IS IT AN EVEN PASS ?
026152  001004                          BNE     2$                      ;IF NOT THEN GO TO 2$
026154  013704  025422                  MOV     @#TEMP2,R4              ;OTHERWISE EXECUTE ASH INSTRUCTION IN MODE 0
026160  072304                          ASH     R4,R3                   ;USING R3
026162  000402                          BR      4$
026164  072367  177232          2$:     ASH     TEMP2,%3                ;SHIFT R3 BY THE NUMBLR SPECIFIED BY TEMP2
026170  106737  025416          4$:     MFPS    @#PSWORD                ;SAVE PS
026174  123737  025426  025416          CMPB    @#TEMP4,@#PSWORD;IS THE PS = TEMP4 ?
026202  001401                          BEQ     11$
026204  104000                          EMT                             ;THE PS IS NOT EQUAL TO 0.
026206  005237  025414          11$:    INC     @#COUNT
026212  023703  025424                  CMP     @#TEMP3,%3              ;IS THE RESULT IN R3 EQUAL TO TEMP3?
```

```
026216  001401                          BEQ     12$
026220                          6$:
026220  104000                          EMT                     ;EITHER INCORRECT R3 OR INCORRECT SEQUENCE
026222  021537  025414          12$:    CMP     (R5),@#COUNT    ;IS THE TEST NUMBER EQUAL TO THE COUNTER?
026226  001374                          BNE     6$              ;IF NOT GO TO THE HLT ABOVE
026230  005215                          INC     (R5)
026232  021527  000037                  CMP     (R5),#37        ;HAS THE CONTENTS OF REGISTERS BEEN SHIFTED
                                                                ;LEFT BY 14. AND RIGHT BY 14.?
026236  002010                          BGE     8$
026240  005237  025422                  INC     @#TEMP2
026244  006367  177154                  ASL     TEMP3           ;SHIFT TEMP3 LEFT?
026250  021527  000020                  CMP     (R5),#20        ;HAS THE CONTENTS OF REGISTERS BEEN SHIFTED LEFT BY 14.?
026254  001003                          BNE     REGR4
026256  000534                          BR      NEGAT           ;IF SO GO TO NEGAT AND INITIATE RIGHT SHIFT
026260  004767  000312          8$:     JSR     PC,TST37        ;IF SO GO AND CONTINUE THE REST OF THE PROGRAM
026264  013704  025420          REGR4:  MOV     @#TEMP1,%4      ;LOAD R4 WITH THE CONTENTS OF TEMP1
026270  010501                          MOV     R5,R1           ;SAVE R5
026272  032737  000001  001006          BIT     #1,@#$PASS      ;IS IT AN EVEN PASS ?
026300  001004                          BNE     2$              ;IF NOT THEN GO TO 2$
026302  013705  025422                  MOV     @#TEMP2,R5      ;OTHERWISE EXECUTE ASH INSTRUCTION IN MODE 0
026306  072405                          ASH     R5,R4           ;USING R4
026310  000402                          BR      4$
026312  072467  177104          2$:     ASH     TEMP2,%4        ;SHIFT R4 BY THE NUMBER SPECIFIED BY TEMP2
026316  106737  025416          4$:     MFPS    @#PSWORD        ;SAVE PS
026322  123737  025426  025416          CMPB    @#TEMP4,@#PSWORD;IS PS = TEMP4 ?
026330  001401                          BEQ     11$
026332  104000                          EMT                     ;THE PS IS NOT EQUAL TO 0
026334  005237  025414          11$:    INC     @#COUNT
026340  023704  025424                  CMP     @#TEMP3,%4      ;IS THE RESULT IN R4 EQUAL TO TEMP3?
026344  001401                          BEQ     12$
026346                          6$:
026346  104000                          EMT                     ;EITHER INCORRECT R4 OR INCORRECT SEQUENCE
026350  010105                  12$:    MOV     R1,R5           ;RESTORE R5
026352  021537  025414                  CMP     (R5),@#COUNT    ;IS THE TEST NUMBER EQUAL TO THE COUNTER?
026356  001373                          BNE     6$              ;IF NOT GO TO THE HLT ABOVE
026360  005215                          INC     (R5)
026362  021527  000037                  CMP     (R5),#37        ;HAS THE CONTENTS OF REGISTERS BEEN
                                                                ;SHIFTED LEFT BY 14. AND RIGHT BY 14.?
026366  002010                          BGE     8$
026370  005237  025422                  INC     @#TEMP2
026374  006367  177024                  ASL     TEMP3           ;SHIFT TEMP3 LEFT
026400  021527  000020                  CMP     (R5),#20        ;HAS THE CONTENTS OF REGISTER BEEN SHIFTED BY 14.?
026404  001003                          BNE     REGR5
026406  000460                          BR      NEGAT           ;IF SO GO TO NEGAT AND INITIATE RIGHT SHIFT
026410  004767  000162          8$:     JSR     PC,TST37        ;IF SO GO AND CONTINUE THE REST OF THE PROGRAM
026414  010501                  REGR5:  MOV     R5,R1           ;SAVE R5
026416  013705  025420                  MOV     @#TEMP1,%5      ;LOAD R5 WITH THE CONTENTS OF TEMP1
026422  032737  000001  001006          BIT     #1,@#$PASS      ;IS IT AN EVEN PASS ?
026430  001004                          BNE     2$              ;IF NOT THEN GO TO 2$
026432  013700  025422                  MOV     @#TEMP2,R0      ;OTHERWISE EXECUTE ASH INSTRUCTION IN MODE 0
026436  072500                          ASH     R0,R5           ;USING R5
026440  000402                          BR      4$
026442  072567  176754          2$:     ASH     TEMP2,%5        ;SHIFT R5 BY THE NUMBER SPECIFIED BY TEMP2
026446  106737  025416          4$:     MFPS    @#PSWORD        ;SAVE PS
026452  123737  025426  025416          CMPB    @#TEMP4,@#PSWORD;IS PS = TEMP4 ?
026460  001401                          BEQ     11$
```

```
026462  104000                                  EMT                             ;THE PS IS NOT EQUAL TO 0.
026464  005237  025414                  11$:    INC     @#COUNT
026470  023705  025424                          CMP     @#TEMP3,%5              ;IS THE RESULT IN R5 EQUAL TO TEMP3?
026474  001401                                  BEQ     12$
026476                                  6$:
026476  104000                                  EMT                             ;EITHER INCORRECT R5 OR INCORRECT SEQUENCE
026500  021137  025414                  12$:    CMP     (R1),@#COUNT            ;IS THE TFST NUMBER EQUAL TO THE COUNTER?
026504  001374                                  BNE     6$                      ;IF NOT GO TO THE HLT ABOVE
026506  010105                                  MOV     R1,R5                   ;RESTORE R5
026510  005215                                  INC     (R5)
026512  021527  000037                          CMP     (R5),#37                ;HAS THE CONTENTS OF REGISTERS BEEN SHIFTED
                                                                                ;LEFT BY 14. AND RIGHT BY 14.?
026516  002010                                  BGE     8$                      ;IF SO GO AND CONTINUE THE REST OF THE PROGRAM
026520  005237  025422                          INC     @#TEMP2
026524  006367  176674                          ASL     TEMP3                   ;SHIFT TEMP3 LEFT
026530  021527  000020                          CMP     (R5),#20                ;HAS THE CONTENTS OF REGISTERS BEEN SHIFTED LEFT BY 14.?
026534  001405                                  BEQ     NEGAT                   ;IF SO GO TO NEGAT AND INITIATE RIGHT SHIFT
026536  000402                                  BR      10$
026540  004767  000032                  8$:     JSR     PC,TST37
026544  000167  176766                  10$:    JMP     ASTART                  ;GO BACK TO START
026550  012737  040000  025420          NEGAT:  MOV     #40000,@#TEMP1          ;TEMP1=40000
026556  012737  177762  025422                  MOV     #177762,@#TEMP2         ;TEMP2=177762
026564  012737  000001  025424                  MOV     #1,@#TEMP3              ;TEMP3=1
026572  000167  176740                          JMP     ASTART
026576  021527  000037                  TST37:  CMP     (R5),#37                ;IS IT TEST 37?
026602  001013                                  BNE     TST40                   ;IF NOT THEN TRY TEST 40
026604  005037  025420                          CLR     @#TEMP1                 ;0
026610  012737  000020  025422                  MOV     #16.,@#TEMP2            ;SHIFTED BY 16
026616  005037  025424                          CLR     @#TEMP3                 ;IS=0
026622  012737  000004  025426                  MOV     #4,@#TEMP4              ;AND PS=4
026630  000207                                  RTS     PC
026632  021527  000040                  TST40:  CMP     (R5),#40                ;IS IT TEST 40?
026636  001003                                  BNE     TST41                   ;IF NOT THEN TRY TEST 41
026640  005037  025422                          CLR     @#TEMP2                 ;0 SHIFTED BY 0=0 AND PS=4
026644  000207                                  RTS     PC
026646  021527  000041                  TST41:  CMP     (R5),#41                ;IS IT TEST 41?
026652  001004                                  BNE     TST42                   ;IF NOT THEN TRY TEST 42
026654  012737  177760  025422                  MOV     #-16.,@#TEMP2           ;0 SHIFTED BY -16.=0 AND PS=4
026662  000207                                  RTS     PC
026664  021527  000042                  TST42:  CMP     (R5),#42                ;IS IT TEST 42?
026670  001013                                  BNE     TST43                   ;IF NOT THEN TRY TEST 43
026672  012737  100000  025420                  MOV     #100000,@#TEMP1         ;100000
026700  005237  025422                          INC     @#TEMP2                 ;SHIFTED BY -15
026704  005337  025424                          DEC     @#TEMP3                 ;IS=-1
026710  012737  000010  025426                  MOV     #10,@#TEMP4                     ;AND PS=10
026716  000207                                  RTS     PC
026720  021527  000043                  TST43:  CMP     (R5),#43                ;IS IT TEST 43?
026724  001012                                  BNE     TST44                   ;IF NOT THEN IF NOT THEN TRY TEST 44
026726  012737  125252  025420                  MOV     #125252,@#TEMP1         ;125252
026734  012737  177777  025422                  MOV     #-1,@#TEMP2             ;SHIFTED BY -1
026742  012737  152525  025424                  MOV     #152525,@#TEMP3         ;IS=152525 AND PS=10
026750  000207                                  RTS     PC
026752  021527  000044                  TST44:  CMP     (R5),#44                ;IS IT TEST 44?
026756  001012                                  BNE     TST45                   ;IF NOT THEN TRY TEST 45
026760  012737  000001  025422                  MOV     #1,@#TEMP2              ;125252 SHIFTED BY 1
026766  012737  052524  025424                  MOV     #52524,@#TEMP3          ;IS=52524
```

```
027774  012737  000003  025426             MOV     #3,@#TEMP4        ;AND PS=3
027002  000207                             RTS     PC
027004  021527  000045          TST45:     CMP     (R5),#45          ;IS IT TEST 45?
027010  001012                             BNE     TST46             ;IF NOT THEN TRY TEST 46
027012  012737  177776  025422             MOV     #-2,@#TEMP2       ;125252 SHIFTED BY -2
027020  012737  165252  025424             MOV     #165252,@#TEMP3   ;IS=165252
027026  012737  000011  025426             MOV     #11,@#TEMP4       ;AND PS=11
027034  000207                             RTS     PC
027036  021527  000046          TST46:     CMP     (R5),#46          ;IS IT TEST 46?
027042  001014                             BNE     TST47             ;IF NOT THEN TRY TEST 47
027044  012737  177777  025420             MOV     #-1,@#TEMP1       ;-1
027052  012737  000020  025422             MOV     #16.,@#TEMP2      ;SHIFTED BY 15.
027060  005037  025424                     CLR     @#TEMP3           ;IS=0
027064  012737  000007  025426             MOV     #7,@#TEMP4        ;AND PS=7
027072  000207                             RTS     PC
027074  021527  000047          TST47:     CMP     (R5),#47          ;IS IT TEST 47?
027100  001011                             BNE     TST50             ;IF NOT THEN TRY TEST 50
027102  005337  025422                     DEC     @#TEMP2           ;-1 SHIFTED BY 15
027106  012737  100000  025424             MOV     #100000,@#TEMP3   ;IS=100000
027114  012737  000011  025426             MOV     #11,@#TEMP4       ;AND PS=11
027122  000207                             RTS     PC
027124  021527  000050          TST50:     CMP     (R5),#50          ;IS IT TEST 50
027130  001007                             BNE     ENT51             ;IF NOT THEN TRY TEST 51
027132  012737  137777  025420             MOV     #137777,@#TEMP1   ;137777 SHIFTED BY 15. IS=100000
027140  012737  000013  025426             MOV     #13,@#TEMP4       ;AND PS=13
027146  000207                             RTS     PC
027150  021527  000051          ENT51:     CMP     (R5),#51          ;IS IT ENTERING TEST 51?
027154  001401                             BEQ     1$
027156  104000                             EMT                       ;TEST NUMBER GOOFED

027160  005726                  1$:        TST     (SP)+             ;RESTORE STACK POINTER
027162  012704  177771                     MOV     #-7,%4
027166  012702  025434                     MOV     #S1,%2
027172  012703  025436                     MOV     #S2,%3
                                ;*****************************************************************
                                ;TEST:51    11/34   ASH     125252 SHIFTED BY #5 = 52500   PS = 3
                                ;*****************************************************************

027176  012701  125252          TST51:     MOV     #125252,%1        ;LOAD R1 WITH 125252
027202  072127  000005                     ASH     #5,%1             ;SHIFT R1 BY #5
027206  106737  025416                     MFPS    @#PSWORD          ;SAVE PS
027212  122737  000003  025416             CMPB    #3,@#PSWORD       ;IS THE PS 3?
027220  001401                             BEQ     11$
027222  104000                             EMT                       ;THE PS IS NOT EQUAL TO 3
027224  022701  052500          11$:       CMP     #52500,%1         ;IS THE RESULT 52500?
027230  001401                             BEQ     12$
027232  104000                             EMT                       ;R1 IS NOT EQUAL TO 52500 OR INCORRECT SEQUENCE
027234  005215                  12$:       INC     (R5)



                                ;*****************************************************************
                                ;TEST:52    11/34   ASH     125252 SHIFTED BY @S2 = 177525   PS = 10
                                ;*****************************************************************

027236  012700  125252          TST52:     MOV     #125252,%0        ;LOAD R0 WITH 125252
```

```
027242 072077 176170                      ASH     @S2,%0          ;SHIFT R0 BY @S2
027246 106737 025416                      MFPS    @#PSWORD        ;SAVE PS
027252 122737 000010 025416               CMPB    #10,@#PSWORD    ;IS THE PS 10?
027260 001401                             BEQ     11$
027262 104000                             EMT                     ;THE PS IS NOT EQUAL TO 10
027264 022700 177525              11$:    CMP     #177525,%0      ;IS THE RESULT 177525?
027270 001401                             BEQ     12$
027272 104000                             EMT                     ;R0 IS NOT EQUAL TO 177525 OR INCORRECT SEQUENCE
027274 005215                     12$:    INC     (R5)



                                  ;*******************************************************************
                                  ;TEST:53   11/34   ASH     125252 SHIFTED BY @#S1 = 177525  PS = 10
                                  ;*******************************************************************

027276 012700 125252              TST53:  MOV     #125252,%0      ;LOAD R0 WITH 125252
027302 072037 025434                      ASH     @#S1,%0         ;SHIFT R0 BY @#S1
027306 106737 025416                      MFPS    @#PSWORD        ;SAVE PS
027312 122737 000010 025416               CMPB    #10,@#PSWORD    ;IS THE PS 10?
027320 001401                             BEQ     11$
027322 104000                             EMT                     ;THE PS IS NOT EQUAL TO 10
027324 022700 177525              11$:    CMP     #177525,%0      ;IS THE RESULT 177525?
027330 001401                             BEQ     12$
027332 104000                             EMT                     ;R0 IS NOT EQUAL TO 177525 OR INCORRECT SEQUENCE
027334 005215                     12$:    INC     (R5)



                                  ;*******************************************************************
                                  ;TEST:54   11/34   ASH     125252 SHIFTED BY (2) = 177525  PS = 10
                                  ;*******************************************************************

027336 012700 125252              TST54:  MOV     #125252,%0      ;LOAD R0 WITH 125252
027342 072012                             ASH     (2),%0          ;SHIFT R0 BY (2)
027344 106737 025416                      MFPS    @#PSWORD        ;SAVE PS
027350 122737 000010 025416               CMPB    #10,@#PSWORD    ;IS THE PS 10?
027356 001401                             BEQ     11$
027360 104000                             EMT                     ;THE PS IS NOT EQUAL TO 10
027362 022700 177525              11$:    CMP     #177525,%0      ;IS THE RESULT 177525?
027366 001401                             BEQ     12$
027370 104000                             EMT                     ;R0 IS NOT EQUAL TO 177525 OR INCORRECT SEQUENCE
027372 005215                     12$:    INC     (R5)



                                  ;*******************************************************************
                                  ;TEST:55   11/34   ASH     125252 SHIFTED BY (2)+ = 177525  PS = 10
                                  ;*******************************************************************

027374 012700 125252              TST55:  MOV     #125252,%0      ;LOAD R0 WITH 125252
027400 072022                             ASH     (2)+,%0         ;SHIFT R0 BY (2)+
027402 106737 025416                      MFPS    @#PSWORD        ;SAVE PS
027406 122737 000010 025416               CMPB    #10,@#PSWORD    ;IS THE PS 10?
027414 001401                             BEQ     11$
027416 104000                             EMT                     ;THE PS IS NOT EQUAL TO 10
```

```
027420  022700  177525          11$:    CMP     #177525,%0      ;IS THE RESULT 177525?
027424  001401                          BEQ     12$
027426  104000                          EMT                     ;RO IS NOT EQUAL TO 177525 OR INCORRECT SEQUENCE
027430  005215                  12$:    INC     (R5)



                                ;*****************************************************************
                                ;TEST:56   11/34   ASH     125252 SHIFTED BY -(2) = 177525  PS = 10
                                ;*****************************************************************

027432  012700  125252          TST56:  MOV     #125252,%0      ;LOAD RO WITH 125252
027436  072042                          ASH     -(2),%0         ;SHIFT RO BY -(2)
027440  106737  025416                  MFPS    @#PSWORD        ;SAVE PS
027444  122737  000010  025416          CMPB    #10,@#PSWORD    ;IS THE PS 10?
027452  001401                          BEQ     11$
027454  104000                          EMT                     ;THE PS IS NOT EQUAL TO 10
027456  022700  177525          11$:    CMP     #177525,%0      ;IS THE RESULT 177525?
027462  001401                          BEQ     12$
027464  104000                          EMT                     ;RO IS NOT EQUAL TO 177525 OR INCORRECT SEQUENCE
027466  005215                  12$:    INC     (R5)



                                ;*****************************************************************
                                ;TEST:57   11/34   ASH     125252 SHIFTED BY 2(3) = 177252  PS = 11
                                ;*****************************************************************

027470  012700  125252          TST57:  MOV     #125252,%0      ;LOAD RO WITH 125252
027474  072063  000002                  ASH     2(3),%0         ;SHIFT RO BY 2(3)
027500  106737  025416                  MFPS    @#PSWORD        ;SAVE PS
027504  122737  000011  025416          CMPB    #11,@#PSWORD    ;IS THE PS 11?
027512  001401                          BEQ     11$
027514  104000                          EMT                     ;THE PS IS NOT EQUAL TO 11
027516  022700  177252          11$:    CMP     #177252,%0      ;IS THE RESULT 177252?
027522  001401                          BEQ     12$
027524  104000                          EMT                     ;RO IS NOT EQUAL TO 177252 OR INCORRECT SEQUENCE
027526  005215                  12$:    INC     (R5)



                                ;*****************************************************************
                                ;TEST:60   11/34   ASH     125252 SHIFTED BY @(3) = 177525  PS = 10
                                ;*****************************************************************

027530  012700  125252          TST60:  MOV     #125252,%0      ;LOAD RO WITH 125252
027534  072073  000000                  ASH     @(3),%0         ;SHIFT RO BY @(3)
027540  106737  025416                  MFPS    @#PSWORD        ;SAVE PS
027544  122737  000010  025416          CMPB    #10,@#PSWORD    ;IS THE PS 10?
027552  001401                          BEQ     11$
027554  104000                          EMT                     ;THE PS IS NOT EQUAL TO 10
027556  022700  177525          11$:    CMP     #177525,%0      ;IS THE RESULT 177525?
027562  001401                          BEQ     12$
027564  104000                          EMT                     ;RO IS NOT EQUAL TO 177525 OR INCORRECT SEQUENCE
027566  005215                  12$:    INC     (R5)
```

```



                                        ;*****************************************************************
                                        ;TEST:61    11/34   ASH     125252 SHIFTED BY @(3)+ = 177525  PS = 10
                                        ;*****************************************************************

027570  012700  125252                  TST61:  MOV     #125252,%0      ;LOAD RO WITH 125252
027574  072033                                  ASH     @(3)+,%0        ;SHIFT RO BY @(3)+
027576  106737  025416                          MFPS    @#PSWORD        ;SAVE PS
027602  122737  000010  025416                  CMPB    #10,@#PSWORD    ;IS THE PS 10?
027610  001401                                  BEQ     11$
027612  104000                                  EMT                     ;THE PS IS NOT EQUAL TO 10
027614  022700  177525                  11$:    CMP     #177525,%0      ;IS THE RESULT 177525?
027620  001401                                  BEQ     12$
027622  104000                                  EMT                     ;RO IS NOT EQUAL TO 177525 OR INCORRECT SEQUENCE
027624  005215                          12$:    INC     (R5)



                                        ;*****************************************************************
                                        ;TEST:62    11/34   ASH     125252 SHIFTED BY @-(3) = 177525  PS = 10
                                        ;*****************************************************************

027626  012700  125252                  TST62:  MOV     #125252,%0      ;LOAD RO WITH 125252
027632  072053                                  ASH     @-(3),%0        ;SHIFT RO BY @-(3)
027634  106737  025416                          MFPS    @#PSWORD        ;SAVE PS
027640  122737  000010  025416                  CMPB    #10,@#PSWORD    ;IS THE PS 10?
027646  001401                                  BEQ     11$
027650  104000                                  EMT                     ;THE PS IS NOT EQUAL TO 10
027652  022700  177525                  11$:    CMP     #177525,%0      ;IS THE RESULT 177525?
027656  001401                                  BEQ     12$
027660  104000                                  EMT                     ;RO IS NOT EQUAL TO 177525 OR INCORRECT SEQUENCE
027662  005215                          12$:    INC     (R5)
```

```
                                                        ;**********************************
                                                        ;      ASHC INSTRUCTION TESTS
                                                        ;**********************************



                                                                   ;*************
                                                                   ;TESTS 63-157
                                                                   ;*************


027664  012737  000062  025414            MOV     #62,@#COUNT
027672  005037  025420                    CLR     @#TEMP1           ;TEMP1=0
027676  012737  000001  025422            MOV     #1,@#TEMP2        ;TEMP2=1
027704  005037  025424                    CLR     @#TEMP3           ;TEMP3=0
027710  005037  025426                    CLR     @#TEMP4           ;TEMP4=0
027714  012737  000001  025430            MOV     #1,@#TEMP5        ;TEMP5=1
027722  005037  025432                    CLR     @#TEMP6           ;0 1 SHIFTED BY 0=0 1, PS=0

027726  010502                    REG01:  MOV     R5,R2             ;SAVE R5
027730  013700  025420                    MOV     @#TEMP1,%0        ;PLACE THE CONTENTS OF TEMP1 IN REGISTER 0
027734  013701  025422                    MOV     @#TEMP2,%0!1      ;PLACE THE CONTENTS OF TEMP2 IN REGISTER 1
027740  000241                            CLC
027742  032737  000001  001006            BIT     #1,@#$PASS        ;IS IT AN EVEN PASS ?
027750  001004                            BNE     2$                ;IF NOT THEN GO TO 2$
027752  013705  025424                    MOV     @#TEMP3,R5        ;OTHERWISE EXECUTE ASHC INSTRUCTION IN MODE 0
027756  073005                            ASHC    R5,R0             ;USING R0
027760  000402                            BR      4$
027762  073067  175436            2$:     ASHC    TEMP3,%0          ;ASHC REGISTER 0 BY THE CONTENTS OF TEMP3
027766  106737  025416            4$:     MFPS    @#PSWORD          ;SAVE PS
027772  123737  025432  025416            CMPB    @#TEMP6,@#PSWORD;COMPARE PS WITH THE CONTENTS OF TEMP6
030000  001401                            BEQ     11$
030002  104000                            EMT                       ;WRONG PS
030004  005237  025414            11$:    INC     @#COUNT
030010  023700  025426                    CMP     @#TEMP4,%0        ;IS THE RESULT IN R0 SAME AS TEMP4?
030014  001401                            BEQ     12$
030016  104000                            EMT                       ;WRONG RESULT IN R0
030020  023701  025430            12$:    CMP     @#TEMP5,%1        ;IS THE RESULT IN R1 SAME AS TEMP5?
                                                                    ;TEMP1 TEMP2 SHIFTED BY TEMP3=TEMP4 TEMP5
                                                                    ;AND PS=TEMP6
030024  001401                            BEQ     13$
030026  104000                            EMT                       ;WRONG RESULT IN R1
030030  010205                    13$:    MOV     R2,R5             ;RESTORE R5
030032  021537  025414                    CMP     (R5),@#COUNT      ;IS TEST NUMBER=COUNTER?
030036  001401                            BEQ     14$
030040  104000                            EMT                       ;NO
030042  005215                    14$:    INC     (R5)
030044  021527  000160                    CMP     (R5),#160         ;HAVE THE FIRST 159 TEST BEEN EXECUTED?
030050  002014                            BGE     6$                ;YES
030052  005237  025424                    INC     @#TEMP3
030056  000241                            CLC
030060  006137  025430                    ROL     @#TEMP5           ;ROTATE TEMP5 LEFT BY 1 PLACE
030064  006137  025426                    ROL     @#TEMP4           ;INTRODUCE CARRY FROM TEMP4 IN TEMP5
030070  021527  000121                    CMP     (R5),#121         ;IS IT TEST 121?
030074  001004                            BNE     REGR23
```

```
030076 004467 000344                 JSR     R4,RITSH          ;IF SO THEN GO AND INITIATE RIGHT SHIFT
030102 004767 000374         6$:     JSR     %7,TST160
030106 013702 025420         REGR23: MOV     @#TEMP1,%2        ;PLACE THE CONTENTS OF TEMP1 IN REGISTER 2
030112 013703 025422                 MOV     @#TEMP2,%2!1      ;PLACE THE CONTENTS OF TEMP2 IN REGISTER 3
030116 000241                        CLC
030120 032737 000001 001006          BIT     #1,@#$PASS        ;IS IT AN EVEN PASS ?
030126 001004                        BNE     2$                ;IF NOT THEN GO TO 2$
030130 013704 025424                 MOV     @#TEMP3,R4        ;OTHERWISE EXECUTE ASHC INSTRUCTION IN MODE 0
030134 073204                        ASHC    R4,R2             ;USING R2
030136 000402                        BR      4$
030140 073267 175260         2$:     ASHC    TEMP3,%2          ;ASHC REGISTER 2 BY THE CONTETNS OF TEMP3
030144 106737 025416         4$:     MFPS    @#PSWORD          ;SAVE PS
030150 123737 025432 025416          CMPB    @#TEMP6,@#PSWORD;COMPARE PS WITH THE CONTENTS OF TEMP6
030156 001401                        BEQ     11$
030160 104000                        EMT                       ;WRONG PS
030162 005237 025414         11$:    INC     @#COUNT
030166 023702 025426                 CMP     @#TEMP4,%2        ;IS THE RESULT IN R2 SAME AS TEMP4?
030172 001401                        BEQ     12$
030174 104000                        EMT                       ;WRONG RESULT IN R2
030176 023703 025430         12$:    CMP     @#TEMP5,%3        ;IS THE RESULT IN R3 SAME AS TEMP5?
                                                               ;TEMP1 TEMP2 SHIFTED BY TEMP3=TEMP4 TEMP5
                                                               ;AND PS=TEMP6
030202 001401                        BEQ     13$
030204 104000                        EMT                       ;WRONG RESULT IN R1
030206 021537 025414         13$:    CMP     (R5),@#COUNT      ;IS TEST NUMBER=COUNTER?
030212 001401                        BEQ     14$
030214 104000                        EMT                       ;NO
030216 005215                14$:    INC     (R5)
030220 021527 000160                 CMP     (R5),#160         ;HAVE THE FIRST 159 TEST BEEN EXECUTED?
030224 002014                        BGE     6$                ;YES
030226 005237 025424                 INC     @#TEMP3
030232 000241                        CLC
030234 006137 025430                 ROL     @#TEMP5           ;ROTATE TEMP5 LEFT BY 1 PLACE
030240 006137 025426                 ROL     @#TEMP4           ;INTRODUCE CARRY FROM TEMP5 IN TEMP4
030244 021527 000121                 CMP     (R5),#121         ;IS IT TEST 121?
030250 001004                        BNE     REG45
030252 004467 000170                 JSR     R4,RITSH          ;IF SO THEN GO AND INITIATE RIGHT SHIFT
030256 004767 000220         6$:     JSR     %7,TST160
030262 010501                REG45:  MOV     R5,R1             ;SAVE R5
030264 013704 025420                 MOV     @#TEMP1,%4        ;PLACE THE CONTENTS OF TEMP1 IN REGISTER 4
030270 013705 025422                 MOV     @#TEMP2,%4!1      ;PLACE THE CONTENTS OF TEMP2 IN REGISTER 5
030274 000241                        CLC
030276 032737 000001 001006          BIT     #1,@#$PASS        ;IS IT AN EVEN PASS ?
030304 001004                        BNE     2$                ;IF NOT THEN GO TO 2$
030306 013700 025424                 MOV     @#TEMP3,R0        ;OTHERWISE EXECUTE ASHC INSTRUCTION IN MODE 0
030312 073400                        ASHC    R0,R4             ;USING R4
030314 000402                        BR      4$
030316 073467 175102         2$:     ASHC    TEMP3,%4          ;ASHC REGISTER 4 BY THE CONTENTS OF TEMP3
030322 106737 025416         4$:     MFPS    @#PSWORD          ;SAVE PS
030326 123737 025432 025416          CMPB    @#TEMP6,@#PSWORD;COMPARE PS WITH THE CONTENTS OF TEMP6
030334 001401                        BEQ     11$
030336 104000                        EMT                       ;WRONG PS
030340 005237 025414         11$:    INC     @#COUNT
030344 023704 025426                 CMP     @#TEMP4,%4        ;IS THE RESULT IN R4 SAME AS TEMP4?
030350 001401                        BEQ     12$
030352 104000                        EMT                       ;WRONG RESULT IN R4
```

```
030354  023705  025430          12$:    CMP     @#TEMP5,%5          ;IS THE RESULT IN R5 SAME AS TEMP5?
                                                                    ;TEMP1 TEMP2 SHIFTED BY TEMP3=TEMP4 TEMP5
                                                                    ;AND PS=TEMP6
030360  001401                          BEQ     13$
030362  104000                          EMT                         ;WRONG RESULT IN R5
030364  021137  025414          13$:    CMP     (R1),@#COUNT        ;IS TEST NUMBER=COUNTER?
030370  001401                          BEQ     14$
030372  104000                          EMT                         ;NO
030374  010105                  14$:    MOV     R1,R5               ;RESTORE R5
030376  005215                          INC     (R5)
030400  021527  000160                  CMP     (R5),#160           ;HAVE THE FIRST 159 TEST BEEN EXECUTED?
030404  002014                          BGE     6$                  ;YES
030406  005237  025424                  INC     @#TEMP3
030412  000241                          CLC
030414  006137  025430                  ROL     @#TEMP5             ;ROTATE TEMP5 LEFT BY 1 PLACE
030420  006137  025426                  ROL     @#TEMP4             ;INTRODUCE CARRY FROM TEMP5 IN TEMP4
030424  021527  000121                  CMP     (R5),#121           ;IS IT TEST 121?
030430  001004                          BNE     8$
030432  004467  000010                  JSR     R4,RITSH            ;IF SO THEN GO AND INITIATE RIGHT SHIFT
030436  004767  000040          6$:     JSR     %7,TST160
030442  000167  177260          8$:     JMP     REG01
030446  022424                  RITSH:  CMP     (R4)+,(R4)+         ;MAKE R4 POINT TO THE NEXT REG TAG
030450  012737  040000  025420          MOV     #40000,@#TEMP1      ;TEMP1=4000
030456  005037  025422                  CLR     @#TEMP2             ;TEMP2=0
030462  012737  177742  025424          MOV     #-30.,@#TEMP3       ;TEMP3=-30
030470  005037  025426                  CLR     @#TEMP4             ;TEMP4=0
030474  005237  025430                  INC     @#TEMP5             ;TEMP5=1
030500  000204                          RTS     R4
030502  021527  000160          TST160: CMP     (R5),#160           ;IS IT TEST 160
030506  001010                          BNE     TST161              ;IF NOT THEN TRY TEST 161
030510  005037  025420                  CLR     @#TEMP1             ;0 0 SHIFTED BY 0
030514  005037  025426                  CLR     @#TEMP4             ;IS EQUAL TO 0 0
030520  012737  000004  025432          MOV     #4,@#TEMP6          ;AND PS=4
030526  000207                          RTS     %7
030530  021527  000161          TST161: CMP     (R5),#161           ;IS IT TEST 161
030534  001004                          BNE     TST162
030536  012737  177746  025424          MOV     #-32,@#TEMP3        ;0 0 SHIFTED BY -32=0 0, PS=4
030544  000207                          RTS     %7
030546  021527  000162          TST162: CMP     (R5),#162           ;IS IT TEST 162
030552  001004                          BNE     TST163              ;IF NOT THEN TRY TEST 163
030554  012737  000032  025424          MOV     #32,@#TEMP3         ;0 0 SHIFTED BY 32=0 0, PS=4
030562  000207                          RTS     %7
030564  021527  000163          TST163: CMP     (R5),#163           ;IS IT TEST 163?
030570  001016                          BNE     TST164              ;IF NOT THEN TRY TEST 164
030572  012737  052525  025420          MOV     #52525,@#TEMP1      ;52525 0
030600  012737  177760  025424          MOV     #-16.,@#TEMP3       ;SHIFTED BY -16.
030606  005037  025426                  CLR     @#TEMP4
030612  012737  052525  025430          MOV     #52525,@#TEMP5      ;IS EQUAL TO 0 52525
030620  005037  025432                  CLR     @#TEMP6             ;AND PS = 0
030624  000207                          RTS     %7
030626  021527  000164          TST164: CMP     (R5),#164           ;IS IT TEST 164?
030632  001014                          BNE     TST165              ;IF NOT THEN TRY TEST 165
030634  012737  125252  025420          MOV     #125252,@#TEMP1     ;125252 0 SHIFTED BY -16.
030642  005337  025426                  DEC     @#TEMP4
030646  012737  125252  025430          MOV     #125252,@#TEMP5     ;IS EQUAL TO -1 125252
030654  012737  000010  025432          MOV     #10,@#TEMP6         ;AND PS=10
```

```
030662  000207                             RTS     %7
030664  021527  000165           TST165:   CMP     (R5),#165          ;IS IT TEST 165?
030670  001007                             BNE     TST166             ;IF NOT THEN TRY TEST 166
030672  012737  177777  025420             MOV     #-1,@#TEMP1        ;-1 0 SHIFTED BY -16
030700  012737  177777  025430             MOV     #-1,@#TEMP5        ;IS EQUAL TO -1 -1, AND PS=10
030706  000207                             RTS     %7
030710  021527  000166           TST166:   CMP     (R5),#166          ;IS IT TEST 166?
030714  001011                             BNE     TST167             ;IF NOT THEN TRY TEST 167
030716  012737  100000  025420             MOV     #100000,@#TEMP1    ;100000 0
030724  012737  177740  025424             MOV     #-32.,@#TEMP3      ;SHIFTED BY -32 IS EQUAL TO -1 -1
030732  005237  025432                     INC     @#TEMP6            ;AND PS=11
030736  000207                             RTS     %7
030740  021527  000167           TST167:   CMP     (R5),#167          ;IS IT TEST 167?
030744  001014                             BNE     TST170             ;IF NOT THEN TRY TEST 170
030746  005037  025420                     CLR     @#TEMP1
030752  005337  025422                     DEC     @#TEMP2            ;0 -1
030756  012737  000020  025424             MOV     #16.,@#TEMP3       ;SHIFTED BY 16.
030764  005037  025430                     CLR     @#TEMP5            ;IS EQUAL TO -1 0
030770  005237  025432                     INC     @#TEMP6            ;AND PS=12
030774  000207                             RTS     %7
030776  021527  000170           TST170:   CMP     (R5),#170          ;IS IT TEST 170?
031002  001007                             BNE     TST171             ;IF NOT THEN TRY TEST 171
031004  012737  125252  025422             MOV     #125252,@#TEMP2    ;0 125252 SHIFTED BY 16
031012  012737  125252  025426             MOV     #125252,@#TEMP4    ;IS EQUAL TO 125252 0, AND PS=12
031020  000207                             RTS     %7
031022  021527  000171           TST171:   CMP     (R5),#171          ;IS IT TEST 171?
031026  001010                             BNE     TST172             ;IF NOT THEN TRY TEST 172
031030  005337  025424                     DEC     @#TEMP3            ;0 125252 SHIFTED BY 15
031034  012737  052525  025426             MOV     #52525,@#TEMP4     ;IS EQUAL TO 52525 0
031042  005037  025432                     CLR     @#TEMP6            ;AND PS=0
031046  000207                             RTS     %7
031050  021527  000172           TST172:   CMP     (R5),#172          ;IS IT TEST 172?
031054  001006                             BNE     TST173             ;IF NOT THEN TRY TEST 173
031056  012737  052525  025422             MOV     #52525,@#TEMP2     ;0 52525
031064  005237  025424                     INC     @#TEMP3            ;SHIFTED BY 16. IS EQUAL TO 52525 0, AND PS=0
031070  000207                             RTS     %7
031072  021527  000173           TST173:   CMP     (R5),#173          ;IS IT TEST 173?
031076  001014                             BNE     TST174             ;IF NOT THEN TRY TEST 174
031100  012737  177777  025422             MOV     #-1,@#TEMP2        ;0 -1
031106  005337  025424                     DEC     @#TEMP3            ;SHIFTED BY 15.
031112  012737  077777  025426             MOV     #77777,@#TEMP4
031120  012737  100000  025430             MOV     #100000,@#TEMP5    ;IS EQUAL TO 77777 100000, AND PS=0
031126  000207                             RTS     %7
031130  021527  000174           TST174:   CMP     (R5),#174          ;IS IT TEST 174?
031134  001013                             BNE     TST175             ;IF NOT THEN TRY TEST 175
031136  012737  100000  025420             MOV     #100000,@#TEMP1
031144  005337  025422                     DEC     @#TEMP2            ;100000 -2 SHIFTED BY 15.
031150  005037  025430                     CLR     @#TEMP5            ;IS EQUAL TO 77777 0
031154  012737  000002  025432             MOV     #2,@#TEMP6         ;AND PS=2
031162  000207                             RTS     %7
031164  021527  000175           TST175:   CMP     (R5),#175          ;IS IT TEST 175?
031170  001015                             BNE     ENT176             ;IF NOT THEN TRY TEST 176
031172  012737  177777  025420             MOV     #-1,@#TEMP1
031200  005037  025422                     CLR     @#TEMP2            ;-1 0
031204  005237  025424                     INC     @#TEMP3            ;SHIFTED BY 16.
031210  005037  025426                     CLR     @#TEMP4            ;IS EQUAL TO 0 0
```

```
031214  012737  000007  025432          MOV     #7,@#TEMP6      ;AND PS=7
031222  000207                          RTS     %7
031224  021527  000176          ENT176: CMP     (R5),#176       ;IS THE PROGRM ENTERING TEST 176?
031230  001401                          BEQ     1$
031232  104000                          EMT                     ;TEST NUMBER GOOFED

031234  005726                  1$:     TST     (SP)+           ;RESTORE STACK POINTER

                                ;*****************************************************************
                                ;TEST:176       1 SHIFTED BY 8. = 400  PS = 0
                                ;*****************************************************************

031236                          TST176:
031236  012701  000000                  MOV     #DUMMY,%1       ;LOAD R1 WITH DUMMY
031242  012701  000001                  MOV     #1,%1!1         ;LOAD R1!1 WITH 1
031246  000241                          CLC
031250  073127  000010                  ASHC    #8.,%1          ;SHIFT R1,R1!1 BY 8.
031254  106737  025416                  MFPS    @#PSWORD        ;SAVE PS
031260  122737  000000  025416          CMPB    #0,@#PSWORD     ;IS THE PS 0?
031266  001401                          BEQ     11$
031270  104000                          EMT                     ;THE PS IS NOT EQUAL TO 0
031272  022701  000400          11$:    CMP     #400,%1         ;IS THE RESULT 400?
031276  001401                          BEQ     13$
031300  104000                          EMT                     ;R1 IS NOT EQUAL TO 400
031302                          13$:
031302  005215                          INC     (R5)


                                ;*****************************************************************
                                ;TEST:177       -1 SHIFTED BY 15. = 100000  PS = 11
                                ;*****************************************************************

031304                          TST177:
031304  012703  000000                  MOV     #DUMMY,%3       ;LOAD R3 WITH DUMMY
031310  012703  177777                  MOV     #-1,%3!1        ;LOAD R3!1 WITH -1
031314  000241                          CLC
031316  073327  000017                  ASHC    #15.,%3         ;SHIFT R3,R3!1 BY 15.
031322  106737  025416                  MFPS    @#PSWORD        ;SAVE PS
031326  122737  000011  025416          CMPB    #11,@#PSWORD    ;IS THE PS 11?
031334  001401                          BEQ     11$
031336  104000                          EMT                     ;THE PS IS NOT EQUAL TO 11
031340  022703  100000          11$:    CMP     #100000,%3      ;IS THE RESULT 100000?
031344  001401                          BEQ     13$
031346  104000                          EMT                     ;R3 IS NOT EQUAL TO 100000
031350                          13$:
031350  005215                          INC     (R5)


                                ;*****************************************************************
                                ;TEST:200       52525 SHIFTED BY 0 = 52525  PS = 0
                                ;*****************************************************************

031352                          TST200:
031352  010501                          MOV     R5,R1           ;SAVE R5
031354  012705  000000                  MOV     #DUMMY,%5       ;LOAD R5 WITH DUMMY
031360  012705  052525                  MOV     #52525,%5!1     ;LOAD R5!1 WITH 52525
```

```
031364  000241                           CLC
031366  073527  000000                   ASHC    #0,%5           ;SHIFT R5,R5!1 BY 0
031372  106737  025416                   MFPS    @#PSWORD        ;SAVE PS
031376  122737  000000  025416           CMPB    #0,@#PSWORD     ;IS THE PS 0?
031404  001401                           BEQ     11$
031406  104000                           EMT                     ;THE PS IS NOT EQUAL TO 0
031410  022705  052525          11$:     CMP     #52525,%5       ;IS THE RESULT 52525?
031414  001401                           BEQ     13$
031416  104000                           EMT                     ;R5 IS NOT EQUAL TO 52525
031420                          13$:
031420  010105                           MOV     R1,R5           ;RESTORE R5
031422  005215                           INC     (R5)


                                ;*************************************************************
                                ;TEST:201        20010 SHIFTED BY -13. = 101  PS = 0
                                ;*************************************************************

031424                          TST201:
031424  012701  000000                   MOV     #DUMMY,%1       ;LOAD R1 WITH DUMMY
031430  012701  020010                   MOV     #20010,%1!1     ;LOAD R1!1 WITH 20010
031434  000241                           CLC
031436  073127  177763                   ASHC    #-13.,%1        ;SHIFT R1,R1!1 BY -13.
031442  106737  025416                   MFPS    @#PSWORD        ;SAVE PS
031446  122737  000000  025416           CMPB    #0,@#PSWORD     ;IS THE PS 0?
031454  001401                           BEQ     11$
031456  104000                           EMT                     ;THE PS IS NOT EQUAL TO 0
031460  022701  000101          11$:     CMP     #101,%1         ;IS THE RESULT 101?
031464  001401                           BEQ     13$
031466  104000                           EMT                     ;R1 IS NOT EQUAL TO 101
031470                          13$:
031470  005215                           INC     (R5)


                                ;*************************************************************
                                ;TEST:202        -1 SHIFTED BY 16. = 0  PS = 11
                                ;*************************************************************

031472                          TST202:
031472  012703  000000                   MOV     #DUMMY,%3       ;LOAD R3 WITH DUMMY
031476  012703  177777                   MOV     #-1,%3!1        ;LOAD R3!1 WITH -1
031502  000241                           CLC
031504  073327  000020                   ASHC    #16.,%3         ;SHIFT R3,R3!1 BY 16.
031510  106737  025416                   MFPS    @#PSWORD        ;SAVE PS
031514  122737  000011  025416           CMPB    #11,@#PSWORD    ;IS THE PS 11?
031522  001401                           BEQ     11$
031524  104000                           EMT                     ;THE PS IS NOT EQUAL TO 11
031526  022703  000000          11$:     CMP     #0,%3           ;IS THE RESULT 0?
031532  001401                           BEQ     13$
031534  104000                           EMT                     ;R3 IS NOT EQUAL TO 0
031536                          13$:
031536  005215                           INC     (R5)


                                ;*************************************************************
                                ;TEST:203        1 SHIFTED BY -1 = 100000  PS = 1
```

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

        031540                          TST203:
        031540  010501                          MOV     R5,R1           ;SAVE R5
        031542  012705  000000                  MOV     #DUMMY,%5       ;LOAD R5 WITH DUMMY
        031546  012705  000001                  MOV     #1,%5!1         ;LOAD R5!1 WITH 1
        031552  000241                          CLC
        031554  073527  177777                  ASHC    #-1,%5          ;SHIFT R5,R5!1 BY -1
        031560  106737  025416                  MFPS    @#PSWORD        ;SAVE PS
        031564  122737  000001  025416          CMPB    #1,@#PSWORD     ;IS THE PS 1?
        031572  001401                          BEQ     11$
        031574  104000                          EMT                     ;THE PS IS NOT EQUAL TO 1
        031576  022705  100000          11$:    CMP     #100000,%5      ;IS THE RESULT 100000?
        031602  001401                          BEQ     13$
        031604  104000                          EMT                     ;R5 IS NOT EQUAL TO 100000
        031606                          13$:
        031606  010105                          MOV     R1,R5           ;RESTORE R5
        031610  005215                          INC     (R5)


                                        ;*******************************************************************
                                        ;TEST:204       125252 SHIFTED BY -16. = 125252  PS = 11
                                        ;*******************************************************************

        031612                          TST204:
        031612  012701  000000                  MOV     #DUMMY,%1       ;LOAD R1 WITH DUMMY
        031616  012701  125252                  MOV     #125252,%1!1    ;LOAD R1!1 WITH 125252
        031622  000241                          CLC
        031624  073127  177760                  ASHC    #-16.,%1        ;SHIFT R1,R1!1 BY -16.
        031630  106737  025416                  MFPS    @#PSWORD        ;SAVE PS
        031634  122737  000011  025416          CMPB    #11,@#PSWORD    ;IS THE PS 11?
        031642  001401                          BEQ     11$
        031644  104000                          EMT                     ;THE PS IS NOT EQUAL TO 11
        031646  022701  125252          11$:    CMP     #125252,%1      ;IS THE RESULT 125252?
        031652  001401                          BEQ     13$
        031654  104000                          EMT                     ;R1 IS NOT EQUAL TO 125252
        031656                          13$:
        031656  005215                          INC     (R5)


                                        ;*******************************************************************
                                        ;TEST:205       125252 125252 SHIFTED BY 21. = 52500 000000  PS = 3
                                        ;*******************************************************************

        031660                          TST205:
        031660  012702  125252                  MOV     #125252,%2      ;LOAD R2 WITH 125252
        031664  012703  125252                  MOV     #125252,%2!1    ;LOAD R2!1 WITH 125252
        031670  000241                          CLC
        031672  073227  000025                  ASHC    #21.,%2         ;SHIFT R2,R2!1 BY 21.
        031676  106737  025416                  MFPS    @#PSWORD        ;SAVE PS
        031702  122737  000003  025416          CMPB    #3,@#PSWORD     ;IS THE PS 3?
        031710  001401                          BEQ     11$
        031712  104000                          EMT                     ;THE PS IS NOT EQUAL TO 3
        031714  022702  052500          11$:    CMP     #52500,%2       ;IS THE RESULT 52500?
        031720  001401                          BEQ     12$
        031722  104000                          EMT                     ;R2 IS NOT EQUAL TO 52500
```

```
031724  022703  000000          12$:    CMP     #000000.%2!1    ;IS THE RESULT 000000?
031730  001401                          BEQ     13$
031732  104000                          EMT                     ;R2!1 IS NOT EQUAL TO 000000
031734                          13$:
031734  005215                          INC     (R5)

031736  012702  177771                  MOV     #-7,%2
031742  012703  025434                  MOV     #S1,%3
031746  012704  025436                  MOV     #S2,%4

                                ;***********************************************************************
                                ;TEST:206       125252 125252 SHIFTED BY S1 = 177525 52525  PS = 10
                                ;***********************************************************************

031752                          TST206:
031752  012700  125252                  MOV     #125252,%0              ;LOAD R0 WITH 125252
031756  012701  125252                  MOV     #125252,%0!1    ;LOAD R0!1 WITH 125252
031762  000241                          CLC
031764  073067  173444                  ASHC    S1,%0           ;SHIFT R0,R0!1 BY S1
031770  106737  025416                  MFPS    @#PSWORD        ;SAVE PS
031774  122737  000010  025416          CMPB    #10,@#PSWORD    ;IS THE PS 10?
032002  001401                          BEQ     11$
032004  104000                          EMT                     ;THE PS IS NOT EQUAL TO 10
032006  022700  177525          11$:    CMP     #177525,%0      ;IS THE RESULT 177525?
032012  001401                          BEQ     12$
032014  104000                          EMT                     ;R0 IS NOT EQUAL TO 177525
032016  022701  052525          12$:    CMP     #52525,%0!1     ;IS THE RESULT 52525?
032022  001401                          BEQ     13$
032024  104000                          EMT                     ;R0!1 IS NOT EQUAL TO 52525 OR INCORRECT SEQUENCE
032026                          13$:
032026  005215                          INC     (R5)

                                ;***********************************************************************
                                ;TEST:207       125252 125252 SHIFTED BY @S2 = 177525 52525  PS = 10
                                ;***********************************************************************

032030                          TST207:
032030  012700  125252                  MOV     #125252,%0              ;LOAD R0 WITH 125252
032034  012701  125252                  MOV     #125252,%0!1    ;LOAD R0!1 WITH 125252
032040  000241                          CLC
032042  073077  173370                  ASHC    @S2,%0          ;SHIFT R0,R0!1 BY @S2
032046  106737  025416                  MFPS    @#PSWORD        ;SAVE PS
032052  122737  000010  025416          CMPB    #10,@#PSWORD    ;IS THE PS 10?
032060  001401                          BEQ     11$
032062  104000                          EMT                     ;THE PS IS NOT EQUAL TO 10
032064  022700  177525          11$:    CMP     #177525,%0      ;IS THE RESULT 177525?
032070  001401                          BEQ     12$
032072  104000                          EMT                     ;R0 IS NOT EQUAL TO 177525
032074  022701  052525          12$:    CMP     #52525,%0!1     ;IS THE RESULT 52525?
032100  001401                          BEQ     13$
032102  104000                          EMT                     ;R0!1 IS NOT EQUAL TO 52525 OR INCORRECT SEQUENCE
032104                          13$:
032104  005215                          INC     (R5)
```

```
                                       ;*****************************************************************
                                       ;TEST:210       125252 125252 SHIFTED BY @#S1 = 177525 52525  PS = 10
                                       ;*****************************************************************

032106                                 TST210:
032106  012700  125252                         MOV     #125252,%0              ;LOAD R0 WITH 125252
032112  012701  125252                         MOV     #125252,%0!1    ;LOAD R0!1 WITH 125252
032116  000241                                 CLC
032120  073037  025434                         ASHC    @#S1,%0         ;SHIFT R0,R0!1 BY @#S1
032124  106737  025416                         MFPS    @#PSWORD        ;SAVE PS
032130  122737  000010  025416                 CMPB    #10,@#PSWORD    ;IS THE PS 10?
032136  001401                                 BEQ     11$
032140  104000                                 EMT                     ;THE PS IS NOT EQUAL TO 10
032142  022700  177525                 11$:    CMP     #177525,%0      ;IS THE RESULT 177525?
032146  001401                                 BEQ     12$
032150  104000                                 EMT                     ;R0 IS NOT EQUAL TO 177525
032152  022701  052525                 12$:    CMP     #52525,%0!1     ;IS THE RESULT 52525?
032156  001401                                 BEQ     13$
032160  104000                                 EMT                     ;R0!1 IS NOT EQUAL TO 52525 OR INCORRECT SEQUENCE
032162                                 13$:
032162  005215                                 INC     (R5)



                                       ;*****************************************************************
                                       ;TEST:211       125252 125252 SHIFTED BY (3) = 177525 52525  PS = 10
                                       ;*****************************************************************

032164                                 TST211:
032164  012700  125252                         MOV     #125252,%0              ;LOAD R0 WITH 125252
032170  012701  125252                         MOV     #125252,%0!1    ;LOAD R0!1 WITH 125252
032174  000241                                 CLC
032176  073013                                 ASHC    (3),%0          ;SHIFT R0,R0!1 BY (3)
032200  106737  025416                         MFPS    @#PSWORD        ;SAVE PS
032204  122737  000010  025416                 CMPB    #10,@#PSWORD    ;IS THE PS 10?
032212  001401                                 BEQ     11$
032214  104000                                 EMT                     ;THE PS IS NOT EQUAL TO 10
032216  022700  177525                 11$:    CMP     #177525,%0      ;IS THE RESULT 177525?
032222  001401                                 BEQ     12$
032224  104000                                 EMT                     ;R0 IS NOT EQUAL TO 177525
032226  022701  052525                 12$:    CMP     #52525,%0!1     ;IS THE RESULT 52525?
032232  001401                                 BEQ     13$
032234  104000                                 EMT                     ;R0!1 IS NOT EQUAL TO 52525 OR INCORRECT SEQUENCE
032236                                 13$:
032236  005215                                 INC     (R5)



                                       ;*****************************************************************
                                       ;TEST:212       125252 125252 SHIFTED BY (3)+ = 177525 52525  PS = 10
                                       ;*****************************************************************

032240                                 TST212:
032240  012700  125252                         MOV     #125252,%0              ;LOAD R0 WITH 125252
032244  012701  125252                         MOV     #125252,%0!1    ;LOAD R0!1 WITH 125252
032250  000241                                 CLC
```

```
032252  073023                          ASHC    (3)+,%0                 ;SHIFT RO,RO!1 BY (3)+
032254  106737  025416                  MFPS    @#PSWORD                ;SAVE PS
032260  122737  000010  025416          CMPB    #10,@#PSWORD            ;IS THE PS 10?
032266  001401                          BEQ     11$
032270  104000                          EMT                             ;THE PS IS NOT EQUAL TO 10
032272  022700  177525          11$:    CMP     #177525,%0              ;IS THE RESULT 177525?
032276  001401                          BEQ     12$
032300  104000                          EMT                             ;RO IS NOT EQUAL TO 177525
032302  022701  052525          12$:    CMP     #52525,%0!1             ;IS THE RESULT 52525?
032306  001401                          BEQ     13$
032310  104000                          EMT                             ;RO!1 IS NOT EQUAL TO 52525 OR INCORRECT SEQUENCE
032312                          13$:
032312  005215                          INC     (R5)


                                ;*******************************************************************
                                ;TEST:213       125252 125252 SHIFTED BY -(3) = 177525 52525  PS = 10
                                ;*******************************************************************

032314                          TST213:
032314  012700  125252                  MOV     #125252,%0                      ;LOAD RO WITH 125252
032320  012701  125252                  MOV     #125252,%0!1            ;LOAD RO!1 WITH 125252
032324  000241                          CLC
032326  073043                          ASHC    -(3),%0                 ;SHIFT RO,RO!1 BY -(3)
032330  106737  025416                  MFPS    @#PSWORD                ;SAVE PS
032334  122737  000010  025416          CMPB    #10,@#PSWORD            ;IS THE PS 10?
032342  001401                          BEQ     11$
032344  104000                          EMT                             ;THE PS IS NOT EQUAL TO 10
032346  022700  177525          11$:    CMP     #177525,%0              ;IS THE RESULT 177525?
032352  001401                          BEQ     12$
032354  104000                          EMT                             ;RO IS NOT EQUAL TO 177525
032356  022701  052525          12$:    CMP     #52525,%0!1             ;IS THE RESULT 52525?
032362  001401                          BEQ     13$
032364  104000                          EMT                             ;RO!1 IS NOT EQUAL TO 52525 OR INCORRECT SEQUENCE
032366                          13$:
032366  005215                          INC     (R5)


                                ;*******************************************************************
                                ;TEST:214       125252 125252 SHIFTED BY 2(4) = 177252 125252  PS = 11
                                ;*******************************************************************

032370                          TST214:
032370  012700  125252                  MOV     #125252,%0                      ;LOAD RO WITH 125252
032374  012701  125252                  MOV     #125252,%0!1            ;LOAD RO!1 WITH 125252
032400  000241                          CLC
032402  073064  000002                  ASHC    2(4),%0                 ;SHIFT RO,RO!1 BY 2(4)
032406  106737  025416                  MFPS    @#PSWORD                ;SAVE PS
032412  122737  000011  025416          CMPB    #11,@#PSWORD            ;IS THE PS 11?
032420  001401                          BEQ     11$
032422  104000                          EMT                             ;THE PS IS NOT EQUAL TO 11
032424  022700  177252          11$:    CMP     #177252,%0              ;IS THE RESULT 177252?
032430  001401                          BEQ     12$
032432  104000                          EMT                             ;RO IS NOT EQUAL TO 177252
032434  022701  125252          12$:    CMP     #125252,%0!1            ;IS THE RESULT 125252?
032440  001401                          BEQ     13$
```

```
032442  104000                          EMT                             ;R0!1 IS NOT EQUAL TO 125252 OR INCORRECT SEQUENCE
032444                          13$:
032444  005215                          INC     (R5)


                                ;*********************************************************************
                                ;TEST:215       125252 125252 SHIFTED BY @(4) = 177525 52525  PS = 10
                                ;*********************************************************************

032446                          TST215:
032446  012700  125252                  MOV     #125252,%0              ;LOAD R0 WITH 125252
032452  012701  125252                  MOV     #125252,%0!1    ;LOAD R0!1 WITH 125252
032456  000241                          CLC
032460  073074  000000                  ASHC    @(4),%0         ;SHIFT R0,R0!1 BY @(4)
032464  106737  025416                  MFPS    @#PSWORD        ;SAVE PS
032470  122737  000010  025416          CMPB    #10,@#PSWORD    ;IS THE PS 10?
032476  001401                          BEQ     11$
032500  104000                          EMT                     ;THE PS IS NOT EQUAL TO 10
032502  022700  177525          11$:    CMP     #177525,%0      ;IS THE RESULT 177525?
032506  001401                          BEQ     12$
032510  104000                          EMT                     ;R0 IS NOT EQUAL TO 177525
032512  022701  052525          12$:    CMP     #52525,%0!1     ;IS THE RESULT 52525?
032516  001401                          BEQ     13$
032520  104000                          EMT                     ;R0!1 IS NOT EQUAL TO 52525 OR INCORRECT SEQUENCE
032522                          13$:
032522  005215                          INC     (R5)


                                ;*********************************************************************
                                ;TEST:216       125252 125252 SHIFTED BY @(4)+ = 177525 52525  PS = 10
                                ;*********************************************************************

032524                          TST216:
032524  012700  125252                  MOV     #125252,%0              ;LOAD R0 WITH 125252
032530  012701  125252                  MOV     #125252,%0!1    ;LOAD R0!1 WITH 125252
032534  000241                          CLC
032536  073034                          ASHC    @(4)+,%0                ;SHIFT R0,R0!1 BY @(4)+
032540  106737  025416                  MFPS    @#PSWORD        ;SAVE PS
032544  122737  000010  025416          CMPB    #10,@#PSWORD    ;IS THE PS 10?
032552  001401                          BEQ     11$
032554  104000                          EMT                     ;THE PS IS NOT EQUAL TO 10
032556  022700  177525          11$:    CMP     #177525,%0      ;IS THE RESULT 177525?
032562  001401                          BEQ     12$
032564  104000                          EMT                     ;R0 IS NOT EQUAL TO 177525
032566  022701  052525          12$:    CMP     #52525,%0!1     ;IS THE RESULT 52525?
032572  001401                          BEQ     13$
032574  104000                          EMT                     ;R0!1 IS NOT EQUAL TO 52525 OR INCORRECT SEQUENCE
032576                          13$:
032576  005215                          INC     (R5)


                                ;*********************************************************************
                                ;TEST:217       125252 125252 SHIFTED BY @-(4) = 177525 52525  PS = 10
                                ;*********************************************************************

032600                          TST217:
```

```
032600 012700 125252                MOV     #125252,%0              ;LOAD R0 WITH 125252
032604 012701 125252                MOV     #125252,%0!1    ;LOAD R0!1 WITH 125252
032610 000241                       CLC
032612 073054                       ASHC    @-(4),%0                ;SHIFT R0,R0!1 BY @-(4)
032614 106737 025416                MFPS    @#PSWORD        ;SAVE PS
032620 122737 000010 025416         CMPB    #10,@#PSWORD    ;IS THE PS 10?
032626 001401                       BEQ     11$
032630 104000                       EMT                     ;THE PS IS NOT EQUAL TO 10
032632 022700 177525        11$:    CMP     #177525,%0      ;IS THE RESULT 177525?
032636 001401                       BEQ     12$
032640 104000                       EMT                     ;R0 IS NOT EQUAL TO 177525
032642 022701 052525        12$:    CMP     #52525,%0!1     ;IS THE RESULT 52525?
032646 001401                       BEQ     13$
032650 104000                       EMT                     ;R0!1 IS NOT EQUAL TO 52525 OR INCORRECT SEQUENCE
032652                      13$:
032652 005215                       INC     (R5)
```

```
                                                ;********************************
                                                ;     MUL INSTRUCTION TESTS
                                                ;********************************




                                        ;*******************************************************************
                                        ;TEST:220       MUL       1 * #0 = 0 0        PS = 4
                                        ;*******************************************************************

032654                                  TST220:
032654  012700  000001                          MOV     #1,%0           ;LOAD MULTIPLICAND WITH 1
032660  070027  000000                          MUL     #0,%0           ;MULTIPLY 1 * #0
032664  106737  025416                          MFPS    @#PSWORD        ;SAVE PS
032670  122737  000004  025416                  CMPB    #4,@#PSWORD     ;IS PS = 4
032676  001401                                  BEQ     11$
032700  104000                                  EMT                     ;PS IS WRONG
032702  022700  000000                  11$:    CMP     #0,%0           ;IS HIGH ORDER = 0
032706  001401                                  BEQ     12$
032710  104000                                  EMT                     ;HIGH ORDER IS WRONG
032712  022701  000000                  12$:    CMP     #0,%0!1         ;IS LOW ORDER = 0
032716  001401                                  BEQ     13$
032720  104000                                  EMT                     ;LOW ORDER IS WRONG OR WRONG SEQUENCE
032722                                  13$:
032722  005215                                  INC     (R5)


                                        ;*******************************************************************
                                        ;TEST:221       MUL       -1 * #1 = -1 -1        PS = 10
                                        ;*******************************************************************

032724                                  TST221:
032724  012700  177777                          MOV     #-1,%0          ;LOAD MULTIPLICAND WITH -1
032730  070027  000001                          MUL     #1,%0           ;MULTIPLY -1 * #1
032734  106737  025416                          MFPS    @#PSWORD        ;SAVE PS
032740  122737  000010  025416                  CMPB    #10,@#PSWORD    ;IS PS = 10
032746  001401                                  BEQ     11$
032750  104000                                  EMT                     ;PS IS WRONG
032752  022700  177777                  11$:    CMP     #-1,%0          ;IS HIGH ORDER = -1
032756  001401                                  BEQ     12$
032760  104000                                  EMT                     ;HIGH ORDER IS WRONG
032762  022701  177777                  12$:    CMP     #-1,%0!1        ;IS LOW ORDER = -1
032766  001401                                  BEQ     13$
032770  104000                                  EMT                     ;LOW ORDER IS WRONG OR WRONG SEQUENCE
032772                                  13$:
032772  005215                                  INC     (R5)


                                        ;*******************************************************************
                                        ;TEST:222       MUL       2 * #2 = 0 4        PS = 0
                                        ;*******************************************************************

032774                                  TST222:
032774  012702  000002                          MOV     #2,%2           ;LOAD MULTIPLICAND WITH 2
033000  070227  000002                          MUL     #2,%2           ;MULTIPLY 2 * #2
```

```
033004  106737  025416                  MFPS    @#PSWORD        ;SAVE PS
033010  122737  000000  025416          CMPB    #0,@#PSWORD     ;IS PS = 0
033016  001401                          BEQ     11$
033020  104000                          EMT                     ;PS IS WRONG
033022  022702  000000          11$:    CMP     #0,%2           ;IS HIGH ORDER = 0
033026  001401                          BEQ     12$
033030  104000                          EMT                     ;HIGH ORDER IS WRONG
033032  022703  000004          12$:    CMP     #4,%2!1         ;IS LOW ORDER = 4
033036  001401                          BEQ     13$
033040  104000                          EMT                     ;LOW ORDER IS WRONG OR WRONG SEQUENCE
033042                          13$:
033042  005215                          INC     (R5)


                                ;*****************************************************************
                                ;TEST:223       MUL     1000 * #200 = 1 0       PS = 1
                                ;*****************************************************************

033044                          TST223:
033044  010501                          MOV     R5,R1           ;SAVE R5
033046  012704  001000                  MOV     #1000,%4        ;LOAD MULTIPLICAND WITH 1000
033052  070427  000200                  MUL     #200,%4         ;MULTIPLY 1000 * #200
033056  106737  025416                  MFPS    @#PSWORD        ;SAVE PS
033062  122737  000001  025416          CMPB    #1,@#PSWORD     ;IS PS = 1
033070  001401                          BEQ     11$
033072  104000                          EMT                     ;PS IS WRONG
033074  022704  000001          11$:    CMP     #1,%4           ;IS HIGH ORDER = 1
033100  001401                          BEQ     12$
033102  104000                          EMT                     ;HIGH ORDER IS WRONG
033104  022705  000000          12$:    CMP     #0,%4!1         ;IS LOW ORDER = 0
033110  001401                          BEQ     13$
033112  104000                          EMT                     ;LOW ORDER IS WRONG OR WRONG SEQUENCE
033114                          13$:
033114  010105                          MOV     R1,R5           ;RESTORE R5
033116  005215                          INC     (R5)


                                ;*****************************************************************
                                ;TEST:224       MUL     2 * #77777 = 0 177776       PS = 1
                                ;*****************************************************************

033120                          TST224:
033120  012700  000002                  MOV     #2,%0           ;LOAD MULTIPLICAND WITH 2
033124  070027  077777                  MUL     #77777,%0       ;MULTIPLY 2 * #77777
033130  106737  025416                  MFPS    @#PSWORD        ;SAVE PS
033134  122737  000001  025416          CMPB    #1,@#PSWORD     ;IS PS = 1
033142  001401                          BEQ     11$
033144  104000                          EMT                     ;PS IS WRONG
033146  022700  000000          11$:    CMP     #0,%0           ;IS HIGH ORDER = 0
033152  001401                          BEQ     12$
033154  104000                          EMT                     ;HIGH ORDER IS WRONG
033156  022701  177776          12$:    CMP     #177776,%0!1    ;IS LOW ORDER = 177776
033162  001401                          BEQ     13$
033164  104000                          EMT                     ;LOW ORDER IS WRONG OR WRONG SEQUENCE
033166                          13$:
033166  005215                          INC     (R5)
```

```


                                                ;*****************************************************************
                                                ;TEST:225        MUL      7777 * #10 = 0 77770        PS = 0
                                                ;*****************************************************************

033170                                          TST225:
033170  012702  007777                                  MOV     #7777,%2                ;LOAD MULTIPLICAND WITH 7777
033174  070227  000010                                  MUL     #10,%2                  ;MULTIPLY 7777 * #10
033200  106737  025416                                  MFPS    @#PSWORD                ;SAVE PS
033204  122737  000000  025416                          CMPB    #0,@#PSWORD             ;IS PS = 0
033212  001401                                          BEQ     11$
033214  104000                                          EMT                             ;PS IS WRONG
033216  022702  000000                          11$:    CMP     #0,%2                   ;IS HIGH ORDER = 0
033222  001401                                          BEQ     12$
033224  104000                                          EMT                             ;HIGH ORDER IS WRONG
033226  022703  077770                          12$:    CMP     #77770,%2!1             ;IS LOW ORDER = 77770
033232  001401                                          BEQ     13$
033234  104000                                          EMT                             ;LOW ORDER IS WRONG OR WRONG SEQUENCE
033236                                          13$:
033236  005215                                          INC     (R5)


                                                ;*****************************************************************
                                                ;TEST:226        MUL      77777 * #77777 = 37777 1        PS = 1
                                                ;*****************************************************************

033240                                          TST226:
033240  010501                                          MOV     R5,R1                   ;SAVE R5
033242  012704  077777                                  MOV     #77777,%4               ;LOAD MULTIPLICAND WITH 77777
033246  070427  077777                                  MUL     #77777,%4               ;MULTIPLY 77777 * #77777
033252  106737  025416                                  MFPS    @#PSWORD                ;SAVE PS
033256  122737  000001  025416                          CMPB    #1,@#PSWORD             ;IS PS = 1
033264  001401                                          BEQ     11$
033266  104000                                          EMT                             ;PS IS WRONG
033270  022704  037777                          11$:    CMP     #37777,%4               ;IS HIGH ORDER = 37777
033274  001401                                          BEQ     12$
033276  104000                                          EMT                             ;HIGH ORDER IS WRONG
033300  022705  000001                          12$:    CMP     #1,%4!1                 ;IS LOW ORDER = 1
033304  001401                                          BEQ     13$
033306  104000                                          EMT                             ;LOW ORDER IS WRONG OR WRONG SEQUENCE
033310                                          13$:
033310  010105                                          MOV     R1,R5                   ;RESTORE R5
033312  005215                                          INC     (R5)


                                                ;*****************************************************************
                                                ;TEST:227        MUL      -1 * #77777 = -1 100001        PS = 10
                                                ;*****************************************************************

033314                                          TST227:
033314  012702  177777                                  MOV     #-1,%2                  ;LOAD MULTIPLICAND WITH -1
033320  070227  077777                                  MUL     #77777,%2               ;MULTIPLY -1 * #77777
033324  106737  025416                                  MFPS    @#PSWORD                ;SAVE PS
033330  122737  000010  025416                          CMPB    #10,@#PSWORD            ;IS PS = 10
033336  001401                                          BEQ     11$
```

```
033340  104000                          EMT                             ;PS IS WRONG
033342  022702  177777          11$:    CMP     #-1,%2                  ;IS HIGH ORDER = -1
033346  001401                          BEQ     12$
033350  104000                          EMT                             ;HIGH ORDER IS WRONG
033352  022703  100001          12$:    CMP     #100001,%2!1            ;IS LOW ORDER = 100001
033356  001401                          BEQ     13$
033360  104000                          EMT                             ;LOW ORDER IS WRONG OR WRONG SEQUENCE
033362                          13$:
033362  005215                          INC     (R5)


                                ;*******************************************************************
                                ;TEST:230       MUL     -2 * #77777 = -1 2          PS = 11
                                ;*******************************************************************

033364                          TST230:
033364  012700  177776                  MOV     #-2,%0                  ;LOAD MULTIPLICAND WITH -2
033370  070027  077777                  MUL     #77777,%0               ;MULTIPLY -2 * #77777
033374  106737  025416                  MFPS    @#PSWORD                ;SAVE PS
033400  122737  000011  025416          CMPB    #11,@#PSWORD            ;IS PS = 11
033406  001401                          BEQ     11$
033410  104000                          EMT                             ;PS IS WRONG
033412  022700  177777          11$:    CMP     #-1,%0                  ;IS HIGH ORDER = -1
033416  001401                          BEQ     12$
033420  104000                          EMT                             ;HIGH ORDER IS WRONG
033422  022701  000002          12$:    CMP     #2,%0!1                 ;IS LOW ORDER = 2
033426  001401                          BEQ     13$
033430  104000                          EMT                             ;LOW ORDER IS WRONG OR WRONG SEQUENCE
033432                          13$:
033432  005215                          INC     (R5)


                                ;*******************************************************************
                                ;TEST:231       MUL     125252 * #2 = -1 52524          PS = 11
                                ;*******************************************************************

033434                          TST231:
033434  012702  125252                  MOV     #125252,%2              ;LOAD MULTIPLICAND WITH 125252
033440  070227  000002                  MUL     #2,%2                   ;MULTIPLY 125252 * #2
033444  106737  025416                  MFPS    @#PSWORD                ;SAVE PS
033450  122737  000011  025416          CMPB    #11,@#PSWORD            ;IS PS = 11
033456  001401                          BEQ     11$
033460  104000                          EMT                             ;PS IS WRONG
033462  022702  177777          11$:    CMP     #-1,%2                  ;IS HIGH ORDER = -1
033466  001401                          BEQ     12$
033470  104000                          EMT                             ;HIGH ORDER IS WRONG
033472  022703  052524          12$:    CMP     #52524,%2!1             ;IS LOW ORDER = 52524
033476  001401                          BEQ     13$
033500  104000                          EMT                             ;LOW ORDER IS WRONG OR WRONG SEQUENCE
033502                          13$:
033502  005215                          INC     (R5)


                                ;*******************************************************************
                                ;TEST:232       MUL     125252 * #40000 = 165252 100000         PS = 11
                                ;*******************************************************************
```

```
033504                          TST232:
033504  010501                          MOV     R5,R1           ;SAVE R5
033506  012704  125252                  MOV     #125252,%4      ;LOAD MULTIPLICAND WITH 125252
033512  070427  040000                  MUL     #40000,%4       ;MULTIPLY 125252 * #40000
033516  106737  025416                  MFPS    @#PSWORD        ;SAVE PS
033522  122737  000011  025416          CMPB    #11,@#PSWORD    ;IS PS = 11
033530  001401                          BEQ     11$
033532  104000                          EMT                     ;PS IS WRONG
033534  022704  165252          11$:    CMP     #165252,%4      ;IS HIGH ORDER = 165252
033540  001401                          BEQ     12$
033542  104000                          EMT                     ;HIGH ORDER IS WRONG
033544  022705  100000          12$:    CMP     #100000,%4!1    ;IS LOW ORDER = 100000
033550  001401                          BEQ     13$
033552  104000                          EMT                     ;LOW ORDER IS WRONG OR WRONG SEQUENCE
033554                          13$:
033554  010105                          MOV     R1,R5           ;RESTORE R5
033556  005215                          INC     (R5)


                                ;*****************************************************************
                                ;TEST:233       MUL     107070 * #107070 = 31222 26100       PS = 1
                                ;*****************************************************************

033560                          TST233:
033560  012700  107070                  MOV     #107070,%0      ;LOAD MULTIPLICAND WITH 107070
033564  070027  107070                  MUL     #107070,%0      ;MULTIPLY 107070 * #107070
033570  106737  025416                  MFPS    @#PSWORD        ;SAVE PS
033574  122737  000001  025416          CMPB    #1,@#PSWORD     ;IS PS = 1
033602  001401                          BEQ     11$
033604  104000                          EMT                     ;PS IS WRONG
033606  022700  031222          11$:    CMP     #31222,%0       ;IS HIGH ORDER = 31222
033612  001401                          BEQ     12$
033614  104000                          EMT                     ;HIGH ORDER IS WRONG
033616  022701  026100          12$:    CMP     #26100,%0!1     ;IS LOW ORDER = 26100
033622  001401                          BEQ     13$
033624  104000                          EMT                     ;LOW ORDER IS WRONG OR WRONG SEQUENCE
033626                          13$:
033626  005215                          INC     (R5)


                                ;*****************************************************************
                                ;TEST:234       MUL     -1 * #1 = -1 -1         PS = 10
                                ;*****************************************************************

033630                          TST234:
033630  012701  177777                  MOV     #-1,%1          ;LOAD MULTIPLICAND WITH -1
033634  070127  000001                  MUL     #1,%1           ;MULTIPLY -1 * #1
033640  106737  025416                  MFPS    @#PSWORD        ;SAVE PS
033644  122737  000010  025416          CMPB    #10,@#PSWORD    ;IS PS = 10
033652  001401                          BEQ     11$
033654  104000                          EMT                     ;PS IS WRONG
033656  022701  177777          11$:    CMP     #-1,%1          ;IS HIGH ORDER = -1
033662  001401                          BEQ     12$
033664  104000                          EMT                     ;HIGH ORDER IS WRONG
033666  022701  177777          12$:    CMP     #-1,%1!1        ;IS LOW ORDER = -1
```

```
033672  001401                                  BEQ     13$
033674  104000                                  EMT                     ;LOW ORDER IS WRONG OR WRONG SEQUENCE
033676                                  13$:
033676  005215                                  INC     (R5)


                                        ;*******************************************************************
                                        ;TEST:235       MUL     -1 * #0 = 0 0           PS = 4
                                        ;*******************************************************************

033700                                  TST235:
033700  012703  177777                          MOV     #-1,%3          ;LOAD MULTIPLICAND WITH -1
033704  070327  000000                          MUL     #0,%3           ;MULTIPLY -1 * #0
033710  106737  025416                          MFPS    @#PSWORD        ;SAVE PS
033714  122737  000004  025416                  CMPB    #4,@#PSWORD     ;IS PS = 4
033722  001401                                  BEQ     11$
033724  104000                                  EMT                     ;PS IS WRONG
033726  022703  000000                  11$:    CMP     #0,%3           ;IS HIGH ORDER = 0
033732  001401                                  BEQ     12$
033734  104000                                  EMT                     ;HIGH ORDER IS WRONG
033736  022703  000000                  12$:    CMP     #0,%3!1         ;IS LOW ORDER = 0
033742  001401                                  BEQ     13$
033744  104000                                  EMT                     ;LOW ORDER IS WRONG OR WRONG SEQUENCE
033746                                  13$:
033746  005215                                  INC     (R5)


                                        ;*******************************************************************
                                        ;TEST:236       MUL     77777 * #100000 = 100000 100000         PS = 11
                                        ;*******************************************************************

033750                                  TST236:
033750  010501                                  MOV     R5,R1           ;SAVE R5
033752  012705  077777                          MOV     #77777,%5       ;LOAD MULTIPLICAND WITH 77777
033756  070527  100000                          MUL     #100000,%5      ;MULTIPLY 77777 * #100000
033762  106737  025416                          MFPS    @#PSWORD        ;SAVE PS
033766  122737  000011  025416                  CMPB    #11,@#PSWORD    ;IS PS = 11
033774  001401                                  BEQ     11$
033776  104000                                  EMT                     ;PS IS WRONG
034000  022705  100000                  11$:    CMP     #100000,%5      ;IS HIGH ORDER = 100000
034004  001401                                  BEQ     12$
034006  104000                                  EMT                     ;HIGH ORDER IS WRONG
034010  022705  100000                  12$:    CMP     #100000,%5!1    ;IS LOW ORDER = 100000
034014  001401                                  BEQ     13$
034016  104000                                  EMT                     ;LOW ORDER IS WRONG OR WRONG SEQUENCE
034020                                  13$:
034020  010105                                  MOV     R1,R5           ;RESTORE R5
034022  005215                                  INC     (R5)


                                        ;*******************************************************************
                                        ;TEST:237       MUL     -1 * #77777 = 100001 100001     PS = 10
                                        ;*******************************************************************

034024                                  TST237:
034024  012701  177777                          MOV     #-1,%1          ;LOAD MULTIPLICAND WITH -1
```

```
034030  070127  077777                  MUL     #77777,%1       ;MULTIPLY -1 * #77777
034034  106737  025416                  MFPS    @#PSWORD        ;SAVE PS
034040  122737  000010  025416          CMPB    #10,@#PSWORD    ;IS PS = 10
034046  001401                          BEQ     11$
034050  104000                          EMT                     ;PS IS WRONG
034052  022701  100001          11$:    CMP     #100001,%1      ;IS HIGH ORDER = 100001
034056  001401                          BEQ     12$
034060  104000                          EMT                     ;HIGH ORDER IS WRONG
034062  022701  100001          12$:    CMP     #100001,%1!1    ;IS LOW ORDER = 100001
034066  001401                          BEQ     13$
034070  104000                          EMT                     ;LOW ORDER IS WRONG OR WRONG SEQUENCE
034072                          13$:
034072  005215                          INC     (R5)


                                ;*****************************************************************
                                ;TEST:240       MUL     77777 * #77777 = 1 1        PS = 1
                                ;*****************************************************************

034074                          TST240:
034074  012703  077777                  MOV     #77777,%3       ;LOAD MULTIPLICAND WITH 77777
034100  070327  077777                  MUL     #77777,%3       ;MULTIPLY 77777 * #77777
034104  106737  025416                  MFPS    @#PSWORD        ;SAVE PS
034110  122737  000001  025416          CMPB    #1,@#PSWORD     ;IS PS = 1
034116  001401                          BEQ     11$
034120  104000                          EMT                     ;PS IS WRONG
034122  022703  000001          11$:    CMP     #1,%3           ;IS HIGH ORDER = 1
034126  001401                          BEQ     12$
034130  104000                          EMT                     ;HIGH ORDER IS WRONG
034132  022703  000001          12$:    CMP     #1,%3!1         ;IS LOW ORDER = 1
034136  001401                          BEQ     13$
034140  104000                          EMT                     ;LOW ORDER IS WRONG OR WRONG SEQUENCE
034142                          13$:
034142  005215                          INC     (R5)


                                ;*****************************************************************
                                ;TEST:241       MUL     2 * #2 = 4 4        PS = 0
                                ;*****************************************************************

034144                          TST241:
034144  010501                          MOV     R5,R1           ;SAVE R5
034146  012705  000002                  MOV     #2,%5           ;LOAD MULTIPLICAND WITH 2
034152  070527  000002                  MUL     #2,%5           ;MULTIPLY 2 * #2
034156  106737  025416                  MFPS    @#PSWORD        ;SAVE PS
034162  122737  000000  025416          CMPB    #0,@#PSWORD     ;IS PS = 0
034170  001401                          BEQ     11$
034172  104000                          EMT                     ;PS IS WRONG
034174  022705  000004          11$:    CMP     #4,%5           ;IS HIGH ORDER = 4
034200  001401                          BEQ     12$
034202  104000                          EMT                     ;HIGH ORDER IS WRONG
034204  022705  000004          12$:    CMP     #4,%5!1         ;IS LOW ORDER = 4
034210  001401                          BEQ     13$
034212  104000                          EMT                     ;LOW ORDER IS WRONG OR WRONG SEQUENCE
034214                          13$:
034214  010105                          MOV     R1,R5           ;RESTORE R5
```

```
034216  005215                                  INC     (R5)


034220  012702  040000                          MOV     #40000,%2
034224  012703  025444                          MOV     #S5,%3
034230  012704  025446                          MOV     #S6,%4

                                        ;*****************************************************************
                                        ;TEST:242        MUL      125252 * S5 = 165252 100000        PS = 11
                                        ;*****************************************************************

034234                                  TST242:
034234  012700  125252                          MOV     #125252,%0          ;LOAD MULTIPLICAND WITH 125252
034240  070067  171200                          MUL     S5,%0               ;MULTIPLY 125252 * S5
034244  106737  025416                          MFPS    @#PSWORD            ;SAVE PS
034250  122737  000011  025416                  CMPB    #11,@#PSWORD        ;IS PS = 11
034256  001401                                  BEQ     11$
034260  104000                                  EMT                         ;PS IS WRONG
034262  022700  165252                  11$:    CMP     #165252,%0          ;IS HIGH ORDER = 165252
034266  001401                                  BEQ     12$
034270  104000                                  EMT                         ;HIGH ORDER IS WRONG
034272  022701  100000                  12$:    CMP     #100000,%0!1        ;IS LOW ORDER = 100000
034276  001401                                  BEQ     13$
034300  104000                                  EMT                         ;LOW ORDER IS WRONG OR WRONG SEQUENCE
034302                                  13$:
034302  005215                                  INC     (R5)


                                        ;*****************************************************************
                                        ;TEST:243        MUL      125252 * @S6 = 165252 100000        PS = 11
                                        ;*****************************************************************

034304                                  TST243:
034304  012700  125252                          MOV     #125252,%0          ;LOAD MULTIPLICAND WITH 125252
034310  070077  171132                          MUL     @S6,%0              ;MULTIPLY 125252 * @S6
034314  106737  025416                          MFPS    @#PSWORD            ;SAVE PS
034320  122737  000011  025416                  CMPB    #11,@#PSWORD        ;IS PS = 11
034326  001401                                  BEQ     11$
034330  104000                                  EMT                         ;PS IS WRONG
034332  022700  165252                  11$:    CMP     #165252,%0          ;IS HIGH ORDER = 165252
034336  001401                                  BEQ     12$
034340  104000                                  EMT                         ;HIGH ORDER IS WRONG
034342  022701  100000                  12$:    CMP     #100000,%0!1        ;IS LOW ORDER = 100000
034346  001401                                  BEQ     13$
034350  104000                                  EMT                         ;LOW ORDER IS WRONG OR WRONG SEQUENCE
034352                                  13$:
034352  005215                                  INC     (R5)


                                        ;*****************************************************************
                                        ;TEST:244        MUL      125252 * @#S5 = 165252 100000       PS = 11
                                        ;*****************************************************************

034354                                  TST244:
034354  012700  125252                          MOV     #125252,%0          ;LOAD MULTIPLICAND WITH 125252
034360  070037  025444                          MUL     @#S5,%0             ;MULTIPLY 125252 * @#S5
```

```
034364  106737  025416                  MFPS    @#PSWORD        ;SAVE PS
034370  122737  000011  025416          CMPB    #11,@#PSWORD    ;IS PS = 11
034376  001401                          BEQ     11$
034400  104000                          EMT                     ;PS IS WRONG
034402  022700  165252          11$:    CMP     #165252,%0      ;IS HIGH ORDER = 165252
034406  001401                          BEQ     12$
034410  104000                          EMT                     ;HIGH ORDER IS WRONG
034412  022701  100000          12$:    CMP     #100000,%0!1    ;IS LOW ORDER = 100000
034416  001401                          BEQ     13$
034420  104000                          EMT                     ;LOW ORDER IS WRONG OR WRONG SEQUENCE
034422                          13$:
034422  005215                          INC     (R5)


                                ;*****************************************************************
                                ;TEST:245       MUL     125252 * %2 = 165252 100000      PS = 11
                                ;*****************************************************************

034424                          TST245:
034424  012700  125252                  MOV     #125252,%0      ;LOAD MULTIPLICAND WITH 125252
034430  070002                          MUL     %2,%0           ;MULTIPLY 125252 * %2
034432  106737  025416                  MFPS    @#PSWORD        ;SAVE PS
034436  122737  000011  025416          CMPB    #11,@#PSWORD    ;IS PS = 11
034444  001401                          BEQ     11$
034446  104000                          EMT                     ;PS IS WRONG
034450  022700  165252          11$:    CMP     #165252,%0      ;IS HIGH ORDER = 165252
034454  001401                          BEQ     12$
034456  104000                          EMT                     ;HIGH ORDER IS WRONG
034460  022701  100000          12$:    CMP     #100000,%0!1    ;IS LOW ORDER = 100000
034464  001401                          BEQ     13$
034466  104000                          EMT                     ;LOW ORDER IS WRONG OR WRONG SEQUENCE
034470                          13$:
034470  005215                          INC     (R5)


                                ;*****************************************************************
                                ;TEST:246       MUL     125252 * (3)+ = 165252 100000      PS = 11
                                ;*****************************************************************

034472                          TST246:
034472  012700  125252                  MOV     #125252,%0      ;LOAD MULTIPLICAND WITH 125252
034476  070023                          MUL     (3)+,%0         ;MULTIPLY 125252 * (3)+
034500  106737  025416                  MFPS    @#PSWORD        ;SAVE PS
034504  122737  000011  025416          CMPB    #11,@#PSWORD    ;IS PS = 11
034512  001401                          BEQ     11$
034514  104000                          EMT                     ;PS IS WRONG
034516  022700  165252          11$:    CMP     #165252,%0      ;IS HIGH ORDER = 165252
034522  001401                          BEQ     12$
034524  104000                          EMT                     ;HIGH ORDER IS WRONG
034526  022701  100000          12$:    CMP     #100000,%0!1    ;IS LOW ORDER = 100000
034532  001401                          BEQ     13$
034534  104000                          EMT                     ;LOW ORDER IS WRONG OR WRONG SEQUENCE
034536                          13$:
034536  005215                          INC     (R5)
```

```
                                        ;*****************************************************************
                                        ;TEST:247      MUL      125252 * -(3) = 165252 100000       PS = 11
                                        ;*****************************************************************

034540                                  TST247:
034540  012700  125252                          MOV     #125252,%0      ;LOAD MULTIPLICAND WITH 125252
034544  070043                                  MUL     -(3),%0         ;MULTIPLY 125252 * -(3)
034546  106737  025416                          MFPS    @#PSWORD        ;SAVE PS
034552  122737  000011  025416                  CMPB    #11,@#PSWORD    ;IS PS = 11
034560  001401                                  BEQ     11$
034562  104000                                  EMT                     ;PS IS WRONG
034564  022700  165252                  11$:    CMP     #165252,%0      ;IS HIGH ORDER = 165252
034570  001401                                  BEQ     12$
034572  104000                                  EMT                     ;HIGH ORDER IS WRONG
034574  022701  100000                  12$:    CMP     #100000,%0!1    ;IS LOW ORDER = 100000
034600  001401                                  BEQ     13$
034602  104000                                  EMT                     ;LOW ORDER IS WRONG OR WRONG SEQUENCE
034604                                  13$:
034604  005215                                  INC     (R5)


                                        ;*****************************************************************
                                        ;TEST:250      MUL      125252 * 2(4) = 165252 100000       PS = 11
                                        ;*****************************************************************

034606                                  TST250:
034606  012700  125252                          MOV     #125252,%0      ;LOAD MULTIPLICAND WITH 125252
034612  070064  000002                          MUL     2(4),%0         ;MULTIPLY 125252 * 2(4)
034616  106737  025416                          MFPS    @#PSWORD        ;SAVE PS
034622  122737  000011  025416                  CMPB    #11,@#PSWORD    ;IS PS = 11
034630  001401                                  BEQ     11$
034632  104000                                  EMT                     ;PS IS WRONG
034634  022700  165252                  11$:    CMP     #165252,%0      ;IS HIGH ORDER = 165252
034640  001401                                  BEQ     12$
034642  104000                                  EMT                     ;HIGH ORDER IS WRONG
034644  022701  100000                  12$:    CMP     #100000,%0!1    ;IS LOW ORDER = 100000
034650  001401                                  BEQ     13$
034652  104000                                  EMT                     ;LOW ORDER IS WRONG OR WRONG SEQUENCE
034654                                  13$:
034654  005215                                  INC     (R5)


                                        ;*****************************************************************
                                        ;TEST:251      MUL      125252 * @(4) = 165252 100000       PS = 11
                                        ;*****************************************************************

034656                                  TST251:
034656  012700  125252                          MOV     #125252,%0      ;LOAD MULTIPLICAND WITH 125252
034662  070074  000000                          MUL     @(4),%0         ;MULTIPLY 125252 * @(4)
034666  106737  025416                          MFPS    @#PSWORD        ;SAVE PS
034672  122737  000011  025416                  CMPB    #11,@#PSWORD    ;IS PS = 11
034700  001401                                  BEQ     11$
034702  104000                                  EMT                     ;PS IS WRONG
034704  022700  165252                  11$:    CMP     #165252,%0      ;IS HIGH ORDER = 165252
034710  001401                                  BEQ     12$
034712  104000                                  EMT                     ;HIGH ORDER IS WRONG
```

```
034714  022701  100000          12$:    CMP     #100000,%0!1    ;IS LOW ORDER = 100000
034720  001401                          BEQ     13$
034722  104000                          EMT                     ;LOW ORDER IS WRONG OR WRONG SEQUENCE
034724                          13$:
034724  005215                          INC     (R5)



                                ;*********************************************************************
                                ;TEST:252       MUL     125252 * @(4)+ = 165252 100000          PS = 11
                                ;*********************************************************************

034726                          TST252:
034726  012700  125252                  MOV     #125252,%0      ;LOAD MULTIPLICAND WITH 125252
034732  070034                          MUL     @(4)+,%0        ;MULTIPLY 125252 * @(4)+
034734  106737  025416                  MFPS    @#PSWORD        ;SAVE PS
034740  122737  000011  025416          CMPB    #11,@#PSWORD    ;IS PS = 11
034746  001401                          BEQ     11$
034750  104000                          EMT                     ;PS IS WRONG
034752  022700  165252          11$:    CMP     #165252,%0      ;IS HIGH ORDER = 165252
034756  001401                          BEQ     12$
034760  104000                          EMT                     ;HIGH ORDER IS WRONG
034762  022701  100000          12$:    CMP     #100000,%0!1    ;IS LOW ORDER = 100000
034766  001401                          BEQ     13$
034770  104000                          EMT                     ;LOW ORDER IS WRONG OR WRONG SEQUENCE
034772                          13$:
034772  005215                          INC     (R5)



                                ;*********************************************************************
                                ;TEST:253       MUL     125252 * @-(4) = 165252 100000          PS = 11
                                ;*********************************************************************

034774                          TST253:
034774  012700  125252                  MOV     #125252,%0      ;LOAD MULTIPLICAND WITH 125252
035000  070054                          MUL     @-(4),%0        ;MULTIPLY 125252 * @-(4)
035002  106737  025416                  MFPS    @#PSWORD        ;SAVE PS
035006  122737  000011  025416          CMPB    #11,@#PSWORD    ;IS PS = 11
035014  001401                          BEQ     11$
035016  104000                          EMT                     ;PS IS WRONG
035020  022700  165252          11$:    CMP     #165252,%0      ;IS HIGH ORDER = 165252
035024  001401                          BEQ     12$
035026  104000                          EMT                     ;HIGH ORDER IS WRONG
035030  022701  100000          12$:    CMP     #100000,%0!1    ;IS LOW ORDER = 100000
035034  001401                          BEQ     13$
035036  104000                          EMT                     ;LOW ORDER IS WRONG OR WRONG SEQUENCE
035040                          13$:
035040  005215                          INC     (R5)
```

```
                                                  ;*******************************
                                                  ;      DIV INSTRUCTION TESTS
                                                  ;*******************************



                                      ;*************************************************************
                                      ;TEST:254       DIV     0 4 / #2 = 2       REM = 0      PS = 0
                                      ;*************************************************************

035042                                TST254:
035042  012700  000000                        MOV     #0,%0           ;LOAD HIGH ORDER WITH 0
035046  012701  000004                        MOV     #4,%0+1         ;LOAD LOW ORDER WITH 4
035052  071027  000002                        DIV     #2,%0           ;DIVIDE BY #2
035056  106737  025416                        MFPS    @#PSWORD        ;SAVE PS
035062  122737  000000  025416                CMPB    #0,@#PSWORD     ;IS PS = 0
035070  001401                                BEQ     11$
035072  104000                                EMT                     ;PS IS WRONG
035074  022700  000002                11$:    CMP     #2,%0           ;IS QUOTIENT = 2
035100  001401                                BEQ     12$
035102  104000                                EMT                     ;QUOTIENT IS WRONG
035104  022701  000000                12$:    CMP     #0,%0+1         ;IS REMAINDER = 0
035110  001401                                BEQ     13$
035112  104000                                EMT                     ;WRONG REMAINDER
035114                                13$:
035114  005215                                INC     (R5)

                                      ;*************************************************************
                                      ;TEST:255       DIV     -1 -9. / #3 = -3       REM = 0      PS = 10
                                      ;*************************************************************

035116                                TST255:
035116  012702  177777                        MOV     #-1,%2          ;LOAD HIGH ORDER WITH -1
035122  012703  177767                        MOV     #-9.,%2+1       ;LOAD LOW ORDER WITH -9.
035126  071227  000003                        DIV     #3,%2           ;DIVIDE BY #3
035132  106737  025416                        MFPS    @#PSWORD        ;SAVE PS
035136  122737  000010  025416                CMPB    #10,@#PSWORD    ;IS PS = 10
035144  001401                                BEQ     11$
035146  104000                                EMT                     ;PS IS WRONG
035150  022702  177775                11$:    CMP     #-3,%2          ;IS QUOTIENT = -3
035154  001401                                BEQ     12$
035156  104000                                EMT                     ;QUOTIENT IS WRONG
035160  022703  000000                12$:    CMP     #0,%2+1         ;IS REMAINDER = 0
035164  001401                                BEQ     13$
035166  104000                                EMT                     ;WRONG REMAINDER
035170                                13$:
035170  005215                                INC     (R5)

                                      ;*************************************************************
                                      ;TEST:256       DIV     0 9. / #2 = 4       REM = 1      PS = 0
                                      ;*************************************************************

035172                                TST256:
035172  010501                                MOV     R5,R1           ;SAVE R5
035174  012704  000000                        MOV     #0,%4           ;LOAD HIGH ORDER WITH 0
035200  012705  000011                        MOV     #9.,%4+1        ;LOAD LOW ORDER WITH 9.
```

```
035204  071427  000002                  DIV     #2,%4               ;DIVIDE BY #2
035210  106737  025416                  MFPS    @#PSWORD            ;SAVE PS
035214  122737  000000  025416          CMPB    #0,@#PSWORD         ;IS PS = 0
035222  001401                          BEQ     11$
035224  104000                          EMT                         ;PS IS WRONG
035226  022704  000004          11$:    CMP     #4,%4               ;IS QUOTIENT = 4
035232  001401                          BEQ     12$
035234  104000                          EMT                         ;QUOTIENT IS WRONG
035236  022705  000001          12$:    CMP     #1,%4+1             ;IS REMAINDER = 1
035242  001401                          BEQ     13$
035244  104000                          EMT                         ;WRONG REMAINDER
035246                          13$:
035246  010105                          MOV     R1,R5               ;RESTORE R5
035250  005215                          INC     (R5)

                                ;*****************************************************************
                                ;TEST:257       DIV     -1 -9. / #2 = -4        REM = -1        PS = 10
                                ;*****************************************************************

035252                          TST257:
035252  012700  177777                  MOV     #-1,%0              ;LOAD HIGH ORDER WITH -1
035256  012701  177767                  MOV     #-9.,%0+1           ;LOAD LOW ORDER WITH -9.
035262  071027  000002                  DIV     #2,%0               ;DIVIDE BY #2
035266  106737  025416                  MFPS    @#PSWORD            ;SAVE PS
035272  122737  000010  025416          CMPB    #10,@#PSWORD        ;IS PS = 10
035300  001401                          BEQ     11$
035302  104000                          EMT                         ;PS IS WRONG
035304  022700  177774          11$:    CMP     #-4,%0              ;IS QUOTIENT = -4
035310  001401                          BEQ     12$
035312  104000                          EMT                         ;QUOTIENT IS WRONG
035314  022701  177777          12$:    CMP     #-1,%0+1            ;IS REMAINDER = -1
035320  001401                          BEQ     13$
035322  104000                          EMT                         ;WRONG REMAINDER
035324                          13$:
035324  005215                          INC     (R5)

                                ;*****************************************************************
                                ;TEST:260       DIV     0 2 / #-3 = 0       REM = 2     PS = 4
                                ;*****************************************************************

035326                          TST260:
035326  012702  000000                  MOV     #0,%2               ;LOAD HIGH ORDER WITH 0
035332  012703  000002                  MOV     #2,%2+1             ;LOAD LOW ORDER WITH 2
035336  071227  177775                  DIV     #-3,%2              ;DIVIDE BY #-3
035342  106737  025416                  MFPS    @#PSWORD            ;SAVE PS
035346  122737  000004  025416          CMPB    #4,@#PSWORD         ;IS PS = 4
035354  001401                          BEQ     11$
035356  104000                          EMT                         ;PS IS WRONG
035360  022702  000000          11$:    CMP     #0,%2               ;IS QUOTIENT = 0
035364  001401                          BEQ     12$
035366  104000                          EMT                         ;QUOTIENT IS WRONG
035370  022703  000002          12$:    CMP     #2,%2+1             ;IS REMAINDER = 2
035374  001401                          BEQ     13$
035376  104000                          EMT                         ;WRONG REMAINDER
035400                          13$:
035400  005215                          INC     (R5)
```

```

                                        ;*****************************************************************
                                        ;TEST:261       DIV     -1 -2 / #3 = 0        REM = -2       PS = 4
                                        ;*****************************************************************

035402                                  TST261:
035402  010501                                  MOV     R5,R1           ;SAVE R5
035404  012704  177777                          MOV     #-1,%4          ;LOAD HIGH ORDER WITH -1
035410  012705  177776                          MOV     #-2,%4+1        ;LOAD LOW ORDER WITH -2
035414  071427  000003                          DIV     #3,%4           ;DIVIDE BY #3
035420  106737  025416                          MFPS    @#PSWORD        ;SAVE PS
035424  122737  000004  025416                  CMPB    #4,@#PSWORD     ;IS PS = 4
035432  001401                                  BEQ     11$
035434  104000                                  EMT                     ;PS IS WRONG
035436  022704  000000                  11$:    CMP     #0,%4           ;IS QUOTIENT = 0
035442  001401                                  BEQ     12$
035444  104000                                  EMT                     ;QUOTIENT IS WRONG
035446  022705  177776                  12$:    CMP     #-2,%4+1        ;IS REMAINDER = -2
035452  001401                                  BEQ     13$
035454  104000                                  EMT                     ;WRONG REMAINDER
035456                                  13$:
035456  010105                                  MOV     R1,R5           ;RESTORE R5
035460  005215                                  INC     (R5)

                                        ;*****************************************************************
                                        ;TEST:262       DIV     -1 -1 / #1 = -1        REM = 0       PS = 10
                                        ;*****************************************************************

035462                                  TST262:
035462  012700  177777                          MOV     #-1,%0          ;LOAD HIGH ORDER WITH -1
035466  012701  177777                          MOV     #-1,%0+1        ;LOAD LOW ORDER WITH -1
035472  071027  000001                          DIV     #1,%0           ;DIVIDE BY #1
035476  106737  025416                          MFPS    @#PSWORD        ;SAVE PS
035502  122737  000010  025416                  CMPB    #10,@#PSWORD    ;IS PS = 10
035510  001401                                  BEQ     11$
035512  104000                                  EMT                     ;PS IS WRONG
035514  022700  177777                  11$:    CMP     #-1,%0          ;IS QUOTIENT = -1
035520  001401                                  BEQ     12$
035522  104000                                  EMT                     ;QUOTIENT IS WRONG
035524  022701  000000                  12$:    CMP     #0,%0+1         ;IS REMAINDER = 0
035530  001401                                  BEQ     13$
035532  104000                                  EMT                     ;WRONG REMAINDER
035534                                  13$:
035534  005215                                  INC     (R5)

                                        ;*****************************************************************
                                        ;TEST:263       DIV     0 0 / #1 = 0        REM = 0       PS = 4
                                        ;*****************************************************************

035536                                  TST263:
035536  012700  000000                          MOV     #0,%0           ;LOAD HIGH ORDER WITH 0
035542  012701  000000                          MOV     #0,%0+1         ;LOAD LOW ORDER WITH 0
035546  071027  000001                          DIV     #1,%0           ;DIVIDE BY #1
035552  106737  025416                          MFPS    @#PSWORD        ;SAVE PS
035556  122737  000004  025416                  CMPB    #4,@#PSWORD     ;IS PS = 4
035564  001401                                  BEQ     11$
```

```
035566 104000                           EMT                     ;PS IS WRONG
035570 022700 000000            11$:    CMP     #0,%0           ;IS QUOTIENT = 0
035574 001401                           BEQ     12$
035576 104000                           EMT                     ;QUOTIENT IS WRONG
035600 022701 000000            12$:    CMP     #0,%0+1         ;IS REMAINDER = 0
035604 001401                           BEQ     13$
035606 104000                           EMT                     ;WRONG REMAINDER
035610                          13$:
035610 005215                           INC     (R5)
                                ;*******************************************************************
                                ;TEST:264       DIV     -1 125252 / #2 = 152525       REM = 0      PS = 10
                                ;*******************************************************************
035612                          TST264:
035612 012702 177777                    MOV     #-1,%2          ;LOAD HIGH ORDER WITH -1
035616 012703 125252                    MOV     #125252,%2+1    ;LOAD LOW ORDER WITH 125252
035622 071227 000002                    DIV     #2,%2           ;DIVIDE BY #2
035626 106737 025416                    MFPS    @#PSWORD        ;SAVE PS
035632 122737 000010 025416             CMPB    #10,@#PSWORD    ;IS PS = 10
035640 001401                           BEQ     11$
035642 104000                           EMT                     ;PS IS WRONG
035644 022702 152525            11$:    CMP     #152525,%2      ;IS QUOTIENT = 152525
035650 001401                           BEQ     12$
035652 104000                           EMT                     ;QUOTIENT IS WRONG
035654 022703 000000            12$:    CMP     #0,%2+1         ;IS REMAINDER = 0
035660 001401                           BEQ     13$
035662 104000                           EMT                     ;WRONG REMAINDER
035664                          13$:
035664 005215                           INC     (R5)
                                ;*******************************************************************
                                ;TEST:265       DIV     -1 -1 / #-1 = 1       REM = 0      PS = 0
                                ;*******************************************************************
035666                          TST265:
035666 010501                           MOV     R5,R1           ;SAVE R5
035670 012704 177777                    MOV     #-1,%4          ;LOAD HIGH ORDER WITH -1
035674 012705 177777                    MOV     #-1,%4+1        ;LOAD LOW ORDER WITH -1
035700 071427 177777                    DIV     #-1,%4          ;DIVIDE BY #-1
035704 106737 025416                    MFPS    @#PSWORD        ;SAVE PS
035710 122737 000000 025416             CMPB    #0,@#PSWORD     ;IS PS = 0
035716 001401                           BEQ     11$
035720 104000                           EMT                     ;PS IS WRONG
035722 022704 000001            11$:    CMP     #1,%4           ;IS QUOTIENT = 1
035726 001401                           BEQ     12$
035730 104000                           EMT                     ;QUOTIENT IS WRONG
035732 022705 000000            12$:    CMP     #0,%4+1         ;IS REMAINDER = 0
035736 001401                           BEQ     13$
035740 104000                           EMT                     ;WRONG REMAINDER
035742                          13$:
035742 010105                           MOV     R1,R5           ;RESTORE R5
035744 005215                           INC     (R5)
                                ;*******************************************************************
                                ;TEST:266       DIV     25253 1 / #125252 = 100000       REM = 1      PS = 10
```

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

        035746                        TST266:
        035746  012700  025253                 MOV     #25253,%0          ;LOAD HIGH ORDER WITH 25253
        035752  012701  000001                 MOV     #1,%0+1            ;LOAD LOW ORDER WITH 1
        035756  071027  125252                 DIV     #125252,%0         ;DIVIDE BY #125252
        035762  106737  025416                 MFPS    @#PSWORD           ;SAVE PS
        035766  122737  000010  025416         CMPB    #10,@#PSWORD       ;IS PS = 10
        035774  001401                         BEQ     11$
        035776  104000                         EMT                        ;PS IS WRONG
        036000  022700  100000        11$:     CMP     #100000,%0         ;IS QUOTIENT = 100000
        036004  001401                         BEQ     12$
        036006  104000                         EMT                        ;QUOTIENT IS WRONG
        036010  022701  000001        12$:     CMP     #1,%0+1            ;IS REMAINDER = 1
        036014  001401                         BEQ     13$
        036016  104000                         EMT                        ;WRONG REMAINDER
        036020                        13$:
        036020  005215                         INC     (R5)

                                      ;*****************************************************************
                                      ;TEST:267     DIV     37777 77777 / #77777 = 77777       REM = 77776       PS = 0
                                      ;*****************************************************************

        036022                        TST267:
        036022  012702  037777                 MOV     #37777,%2          ;LOAD HIGH ORDER WITH 37777
        036026  012703  077777                 MOV     #77777,%2+1        ;LOAD LOW ORDER WITH 77777
        036032  071227  077777                 DIV     #77777,%2          ;DIVIDE BY #77777
        036036  106737  025416                 MFPS    @#PSWORD           ;SAVE PS
        036042  122737  000000  025416         CMPB    #0,@#PSWORD        ;IS PS = 0
        036050  001401                         BEQ     11$
        036052  104000                         EMT                        ;PS IS WRONG
        036054  022702  077777        11$:     CMP     #77777,%2          ;IS QUOTIENT = 77777
        036060  001401                         BEQ     12$
        036062  104000                         EMT                        ;QUOTIENT IS WRONG
        036064  022703  077776        12$:     CMP     #77776,%2+1        ;IS REMAINDER = 77776
        036070  001401                         BEQ     13$
        036072  104000                         EMT                        ;WRONG REMAINDER
        036074                        13$:
        036074  005215                         INC     (R5)

                                      ;*****************************************************************
                                      ;TEST:270     DIV     0 100000 / #2 = 40000       REM = 0      PS = 0
                                      ;*****************************************************************

        036076                        TST270:
        036076  010501                         MOV     R5,R1              ;SAVE R5
        036100  012704  000000                 MOV     #0,%4              ;LOAD HIGH ORDER WITH 0
        036104  012705  100000                 MOV     #100000,%4+1       ;LOAD LOW ORDER WITH 100000
        036110  071427  000002                 DIV     #2,%4              ;DIVIDE BY #2
        036114  106737  025416                 MFPS    @#PSWORD           ;SAVE PS
        036120  122737  000000  025416         CMPB    #0,@#PSWORD        ;IS PS = 0
        036126  001401                         BEQ     11$
        036130  104000                         EMT                        ;PS IS WRONG
        036132  022704  040000        11$:     CMP     #40000,%4          ;IS QUOTIENT = 40000
        036136  001401                         BEQ     12$
        036140  104000                         EMT                        ;QUOTIENT IS WRONG
```

```
036142  022705  000000          12$:    CMP     #0,%4+1         ;IS REMAINDER = 0
036146  001401                          BEQ     13$
036150  104000                          EMT                     ;WRONG REMAINDER
036152                          13$:
036152  010105                          MOV     R1,R5           ;RESTORE R5
036154  005215                          INC     (R5)

                                ;*******************************************************************
                                ;TEST:271      DIV     177777 77777 / #177776 = 40000       REM = 177777      PS = 0
                                ;*******************************************************************

036156                          TST271:
036156  012700  177777                  MOV     #177777,%0      ;LOAD HIGH ORDER WITH 177777
036162  012701  077777                  MOV     #77777,%0+1     ;LOAD LOW ORDER WITH 77777
036166  071027  177776                  DIV     #177776,%0      ;DIVIDE BY #177776
036172  106737  025416                  MFPS    @#PSWORD        ;SAVE PS
036176  122737  000000  025416          CMPB    #0,@#PSWORD     ;IS PS = 0
036204  001401                          BEQ     11$
036206  104000                          EMT                     ;PS IS WRONG
036210  022700  040000          11$:    CMP     #40000,%0       ;IS QUOTIENT = 40000
036214  001401                          BEQ     12$
036216  104000                          EMT                     ;QUOTIENT IS WRONG
036220  022701  177777          12$:    CMP     #177777,%0+1    ;IS REMAINDER = 177777
036224  001401                          BEQ     13$
036226  104000                          EMT                     ;WRONG REMAINDER
036230                          13$:
036230  005215                          INC     (R5)

                                ;*******************************************************************
                                ;TEST:272      DIV     0 52525 / #52525 = 1       REM = 0      PS = 0
                                ;*******************************************************************

036232                          TST272:
036232  012702  000000                  MOV     #0,%2           ;LOAD HIGH ORDER WITH 0
036236  012703  052525                  MOV     #52525,%2+1     ;LOAD LOW ORDER WITH 52525
036242  071227  052525                  DIV     #52525,%2       ;DIVIDE BY #52525
036246  106737  025416                  MFPS    @#PSWORD        ;SAVE PS
036252  122737  000000  025416          CMPB    #0,@#PSWORD     ;IS PS = 0
036260  001401                          BEQ     11$
036262  104000                          EMT                     ;PS IS WRONG
036264  022702  000001          11$:    CMP     #1,%2           ;IS QUOTIENT = 1
036270  001401                          BEQ     12$
036272  104000                          EMT                     ;QUOTIENT IS WRONG
036274  022703  000000          12$:    CMP     #0,%2+1         ;IS REMAINDER = 0
036300  001401                          BEQ     13$
036302  104000                          EMT                     ;WRONG REMAINDER
036304                          13$:
036304  005215                          INC     (R5)

                                ;*******************************************************************
                                ;TEST:273      DIV     0 77777 / #0 = DUMMY       REM = DUMMY      PS = 3
                                ;*******************************************************************

036306                          TST273:
036306  010501                          MOV     R5,R1           ;SAVE R5
036310  012704  000000                  MOV     #0,%4           ;LOAD HIGH ORDER WITH 0
```

```
036314  012705  077777                  MOV     #77777,%4+1     ;LOAD LOW ORDER WITH 77777
036320  071427  000000                  DIV     #0,%4           ;DIVIDE BY #0
036324  106737  025416                  MFPS    @#PSWORD        ;SAVE PS
036330  042737  000014  025416          BIC     #14,@#PSWORD
036336  122737  000003  025416          CMPB    #3,@#PSWORD     ;IS PS = 3
036344  001401                          BEQ     13$
036346  104000                          EMT                     ;PS IS WRONG
036350                          13$:
036350  010105                          MOV     R1,R5           ;RESTORE R5
036352  005215                          INC     (R5)

                                ;*******************************************************************
                                ;TEST:274       DIV     77777 177777 / #2 = DUMMY       REM = DUMMY         PS = 2
                                ;*******************************************************************

036354                          TST274:
036354  012700  077777                  MOV     #77777,%0       ;LOAD HIGH ORDER WITH 77777
036360  012701  177777                  MOV     #177777,%0+1    ;LOAD LOW ORDER WITH 177777
036364  071027  000002                  DIV     #2,%0           ;DIVIDE BY #2
036370  106737  025416                  MFPS    @#PSWORD        ;SAVE PS
036374  042737  000014  025416          BIC     #14,@#PSWORD
036402  122737  000002  025416          CMPB    #2,@#PSWORD     ;IS PS = 2
036410  001401                          BEQ     13$
036412  104000                          EMT                     ;PS IS WRONG
036414                          13$:
036414  005215                          INC     (R5)

036416  012702  000002                  MOV     #2,%2
036422  012703  025454                  MOV     #S9,%3
036426  012704  025456                  MOV     #S10,%4

                                ;*******************************************************************
                                ;TEST:275       DIV     0 52525 / S9 = 25252        REM = 1       PS = 0
                                ;*******************************************************************

036432                          TST275:
036432  012700  000000                  MOV     #0,%0           ;LOAD HIGH ORDER WITH 0
036436  012701  052525                  MOV     #52525,%0+1     ;LOAD LOW ORDER WITH 52525
036442  071067  167006                  DIV     S9,%0           ;DIVIDE BY S9
036446  106737  025416                  MFPS    @#PSWORD        ;SAVE PS
036452  122737  000000  025416          CMPB    #0,@#PSWORD     ;IS PS = 0
036460  001401                          BEQ     11$
036462  104000                          EMT                     ;PS IS WRONG
036464  022700  025252          11$:    CMP     #25252,%0       ;IS QUOTIENT = 25252
036470  001401                          BEQ     12$
036472  104000                          EMT                     ;QUOTIENT IS WRONG
036474  022701  000001          12$:    CMP     #1,%0+1         ;IS REMAINDER = 1
036500  001401                          BEQ     13$
036502  104000                          EMT                     ;WRONG REMAINDER
036504                          13$:
036504  005215                          INC     (R5)

                                ;*******************************************************************
                                ;TEST:276       DIV     0 52525 / @S10 = 25252        REM = 1       PS = 0
                                ;*******************************************************************
```

```
036506                            TST276:
036506  012700  000000                    MOV     #0,%0            ;LOAD HIGH ORDER WITH 0
036512  012701  052525                    MOV     #52525,%0+1      ;LOAD LOW ORDER WITH 52525
036516  071077  166734                    DIV     @S10,%0          ;DIVIDE BY @S10
036522  106737  025416                    MFPS    @#PSWORD         ;SAVE PS
036526  122737  000000  025416            CMPB    #0,@#PSWORD      ;IS PS = 0
036534  001401                            BEQ     11$
036536  104000                            EMT                      ;PS IS WRONG
036540  022700  025252            11$:    CMP     #25252,%0        ;IS QUOTIENT = 25252
036544  001401                            BEQ     12$
036546  104000                            EMT                      ;QUOTIENT IS WRONG
036550  022701  000001            12$:    CMP     #1,%0+1          ;IS REMAINDER = 1
036554  001401                            BEQ     13$
036556  104000                            EMT                      ;WRONG REMAINDER
036560                            13$:
036560  005215                            INC     (R5)

                                  ;*****************************************************************
                                  ;TEST:277     DIV     0 52525 / @#S9 = 25252      REM = 1      PS = 0
                                  ;*****************************************************************

036562                            TST277:
036562  012700  000000                    MOV     #0,%0            ;LOAD HIGH ORDER WITH 0
036566  012701  052525                    MOV     #52525,%0+1      ;LOAD LOW ORDER WITH 52525
036572  071037  025454                    DIV     @#S9,%0          ;DIVIDE BY @#S9
036576  106737  025416                    MFPS    @#PSWORD         ;SAVE PS
036602  122737  000000  025416            CMPB    #0,@#PSWORD      ;IS PS = 0
036610  001401                            BEQ     11$
036612  104000                            EMT                      ;PS IS WRONG
036614  022700  025252            11$:    CMP     #25252,%0        ;IS QUOTIENT = 25252
036620  001401                            BEQ     12$
036622  104000                            EMT                      ;QUOTIENT IS WRONG
036624  022701  000001            12$:    CMP     #1,%0+1          ;IS REMAINDER = 1
036630  001401                            BEQ     13$
036632  104000                            EMT                      ;WRONG REMAINDER
036634                            13$:
036634  005215                            INC     (R5)

                                  ;*****************************************************************
                                  ;TEST:300     DIV     0 52525 / %2 = 25252      REM = 1      PS = 0
                                  ;*****************************************************************

036636                            TST300:
036636  012700  000000                    MOV     #0,%0            ;LOAD HIGH ORDER WITH 0
036642  012701  052525                    MOV     #52525,%0+1      ;LOAD LOW ORDER WITH 52525
036646  071002                            DIV     %2,%0            ;DIVIDE BY %2
036650  106737  025416                    MFPS    @#PSWORD         ;SAVE PS
036654  122737  000000  025416            CMPB    #0,@#PSWORD      ;IS PS = 0
036662  001401                            BEQ     11$
036664  104000                            EMT                      ;PS IS WRONG
036666  022700  025252            11$:    CMP     #25252,%0        ;IS QUOTIENT = 25252
036672  001401                            BEQ     12$
036674  104000                            EMT                      ;QUOTIENT IS WRONG
036676  022701  000001            12$:    CMP     #1,%0+1          ;IS REMAINDER = 1
036702  001401                            BEQ     13$
036704  104000                            EMT                      ;WRONG REMAINDER
```

```
036706                          13$:
036706  005215                          INC     (R5)

                                ;*********************************************************************
                                ;TEST:301       DIV     0 52525 / (3)+ = 25252       REM = 1      PS = 0
                                ;*********************************************************************

036710                          TST301:
036710  012700  000000                  MOV     #0,%0           ;LOAD HIGH ORDER WITH 0
036714  012701  052525                  MOV     #52525,%0+1     ;LOAD LOW ORDER WITH 52525
036720  071023                          DIV     (3)+,%0         ;DIVIDE BY (3)+
036722  106737  025416                  MFPS    @#PSWORD        ;SAVE PS
036726  122737  000000  025416          CMPB    #0,@#PSWORD     ;IS PS = 0
036734  001401                          BEQ     11$
036736  104000                          EMT                     ;PS IS WRONG
036740  022700  025252          11$:    CMP     #25252,%0       ;IS QUOTIENT = 25252
036744  001401                          BEQ     12$
036746  104000                          EMT                     ;QUOTIENT IS WRONG
036750  022701  000001          12$:    CMP     #1,%0+1         ;IS REMAINDER = 1
036754  001401                          BEQ     13$
036756  104000                          EMT                     ;WRONG REMAINDER
036760                          13$:
036760  005215                          INC     (R5)

                                ;*********************************************************************
                                ;TEST:302       DIV     0 52525 / -(3) = 25252       REM = 1      PS = 0
                                ;*********************************************************************

036762                          TST302:
036762  012700  000000                  MOV     #0,%0           ;LOAD HIGH ORDER WITH 0
036766  012701  052525                  MOV     #52525,%0+1     ;LOAD LOW ORDER WITH 52525
036772  071043                          DIV     -(3),%0         ;DIVIDE BY -(3)
036774  106737  025416                  MFPS    @#PSWORD        ;SAVE PS
037000  122737  000000  025416          CMPB    #0,@#PSWORD     ;IS PS = 0
037006  001401                          BEQ     11$
037010  104000                          EMT                     ;PS IS WRONG
037012  022700  025252          11$:    CMP     #25252,%0       ;IS QUOTIENT = 25252
037016  001401                          BEQ     12$
037020  104000                          EMT                     ;QUOTIENT IS WRONG
037022  022701  000001          12$:    CMP     #1,%0+1         ;IS REMAINDER = 1
037026  001401                          BEQ     13$
037030  104000                          EMT                     ;WRONG REMAINDER
037032                          13$:
037032  005215                          INC     (R5)

                                ;*********************************************************************
                                ;TEST:303       DIV     0 52525 / 2(4) = 25252       REM = 1      PS = 0
                                ;*********************************************************************

037034                          TST303:
037034  012700  000000                  MOV     #0,%0           ;LOAD HIGH ORDER WITH 0
037040  012701  052525                  MOV     #52525,%0+1     ;LOAD LOW ORDER WITH 52525
037044  071064  000002                  DIV     2(4),%0         ;DIVIDE BY 2(4)
037050  106737  025416                  MFPS    @#PSWORD        ;SAVE PS
037054  122737  000000  025416          CMPB    #0,@#PSWORD     ;IS PS = 0
037062  001401                          BEQ     11$
```

```
037064  104000                          EMT                     ;PS IS WRONG
037066  022700  025252          11$:    CMP     #25252,%0       ;IS QUOTIENT = 25252
037072  001401                          BEQ     12$
037074  104000                          EMT                     ;QUOTIENT IS WRONG
037076  022701  000001          12$:    CMP     #1,%0+1         ;IS REMAINDER = 1
037102  001401                          BEQ     13$
037104  104000                          EMT                     ;WRONG REMAINDER
037106                          13$:
037106  005215                          INC     (R5)

                                ;*****************************************************************
                                ;TEST:304       DIV     0 52525 / @(4) = 25252        REM = 1      PS = 0
                                ;*****************************************************************

037110                          TST304:
037110  012700  000000                  MOV     #0,%0           ;LOAD HIGH ORDER WITH 0
037114  012701  052525                  MOV     #52525,%0+1     ;LOAD LOW ORDER WITH 52525
037120  071074  000000                  DIV     @(4),%0         ;DIVIDE BY @(4)
037124  106737  025416                  MFPS    @#PSWORD        ;SAVE PS
037130  122737  000000  025416          CMPB    #0,@#PSWORD     ;IS PS = 0
037136  001401                          BEQ     11$
037140  104000                          EMT                     ;PS IS WRONG
037142  022700  025252          11$:    CMP     #25252,%0       ;IS QUOTIENT = 25252
037146  001401                          BEQ     12$
037150  104000                          EMT                     ;QUOTIENT IS WRONG
037152  022701  000001          12$:    CMP     #1,%0+1         ;IS REMAINDER = 1
037156  001401                          BEQ     13$
037160  104000                          EMT                     ;WRONG REMAINDER
037162                          13$:
037162  005215                          INC     (R5)

                                ;*****************************************************************
                                ;TEST:305       DIV     0 52525 / @(4)+ = 25252        REM = 1      PS = 0
                                ;*****************************************************************

037164                          TST305:
037164  012700  000000                  MOV     #0,%0           ;LOAD HIGH ORDER WITH 0
037170  012701  052525                  MOV     #52525,%0+1     ;LOAD LOW ORDER WITH 52525
037174  071034                          DIV     @(4)+,%0        ;DIVIDE BY @(4)+
037176  106737  025416                  MFPS    @#PSWORD        ;SAVE PS
037202  122737  000000  025416          CMPB    #0,@#PSWORD     ;IS PS = 0
037210  001401                          BEQ     11$
037212  104000                          EMT                     ;PS IS WRONG
037214  022700  025252          11$:    CMP     #25252,%0       ;IS QUOTIENT = 25252
037220  001401                          BEQ     12$
037222  104000                          EMT                     ;QUOTIENT IS WRONG
037224  022701  000001          12$:    CMP     #1,%0+1         ;IS REMAINDER = 1
037230  001401                          BEQ     13$
037232  104000                          EMT                     ;WRONG REMAINDER
037234                          13$:
037234  005215                          INC     (R5)

                                ;*****************************************************************
                                ;TEST:306       DIV     0 52525 / @-(4) = 25252        REM = 1      PS = 0
                                ;*****************************************************************
```

```
037236                                  TST306:
037236  012700  000000                          MOV     #0,%0           ;LOAD HIGH ORDER WITH 0
037242  012701  052525                          MOV     #52525,%0+1     ;LOAD LOW ORDER WITH 52525
037246  071054                                  DIV     @-(4),%0        ;DIVIDE BY @-(4)
037250  106737  025416                          MFPS    @#PSWORD        ;SAVE PS
037254  122737  000000  025416                  CMPB    #0,@#PSWORD     ;IS PS = 0
037262  001401                                  BEQ     11$
037264  104000                                  EMT                     ;PS IS WRONG
037266  022700  025252                  11$:    CMP     #25252,%0       ;IS QUOTIENT = 25252
037272  001401                                  BEQ     12$
037274  104000                                  EMT                     ;QUOTIENT IS WRONG
037276  022701  000001                  12$:    CMP     #1,%0+1         ;IS REMAINDER = 1
037302  001401                                  BEQ     13$
037304  104000                                  EMT                     ;WRONG REMAINDER
037306                                  13$:
037306  005215                                  INC     (R5)



                                        ;SPECIAL MULTIPLY DATA PATTERN TEST

037310  012701  177777                  TSTSPC: MOV     #-1,R1          ;MAKE R1 -1 SO WE KNOW INSTR. WAS MODIFIER
037314  012700  077700                          MOV     #77700,R0       ;SET UP TEST DATA
037320  070027  000001                          MUL     #1,R0           ;DO MULTIPLY INSTRUCTION
037324  022701  077700                          CMP     #77700,R1       ;CHECK LOW ORDER WORD
037330  001401                                  BEQ     1$
037332  104000                                  EMT                     ;LOW ORDER PRODUCT ERROR
037334  005700                          1$:     TST     R0              ;CHECK HIGH ORDER WORD
037336  001401                                  BEQ     EISEND
037340  104000                                  EMT                     ;HIGH ORDER PRODUCT ERROR
037342  000167  000026                  EISEND: JMP     MMUTST          ;JMP OVER GARBAGE AND GET TO MMU TEST
```

```

                        .SBTTL  MEMORY MANAGEMENT DEFINITIONS

                        ;*KT11 VECTOR ADDRESS

        000250          MMVEC=  250

                        ;*KT11 STATUS REGISTER ADDRESSES

        177572          SR0=    177572
        177574          SR1=    177574
        177576          SR2=    177576
        172516          SR3=    172516

                        ;*USER "I" PAGE DESCRIPTOR REGISTERS

        177600          UIPDR0= 177600
        177602          UIPDR1= 177602
        177604          UIPDR2= 177604
        177606          UIPDR3= 177606
        177610          UIPDR4= 177610
        177612          UIPDR5= 177612
        177614          UIPDR6= 177614
        177616          UIPDR7= 177616

                        ;*USER "I" PAGE ADDRESS REGISTERS

        177640          UIPAR0= 177640
        177642          UIPAR1= 177642
        177644          UIPAR2= 177644
        177646          UIPAR3= 177646
        177650          UIPAR4= 177650
        177652          UIPAR5= 177652
        177654          UIPAR6= 177654
        177656          UIPAR7= 177656

                        ;*KERNEL "I" PAGE DESCRIPTOR REGISTERS

        172300          KIPDR0= 172300
        172302          KIPDR1= 172302
        172304          KIPDR2= 172304
        172306          KIPDR3= 172306
        172310          KIPDR4= 172310
        172312          KIPDR5= 172312
        172314          KIPDR6= 172314
        172316          KIPDR7= 172316

                        ;*KERNEL "I" PAGE ADDRESS REGISTERS

        172340          KIPAR0= 172340
        172342          KIPAR1= 172342
        172344          KIPAR2= 172344
        172346          KIPAR3= 172346
        172350          KIPAR4= 172350
        172352          KIPAR5= 172352
        172354          KIPAR6= 172354
```

```
         172356         KIPAR7= 172356

         000006         KSP=    SP
         000006         USP=    SP
         177776         PSW=    PS
         000020         TBIT=   20
         000100         WBIT=   100
         000001         BIT0=   1
         000004         ERRVEC= 4

                        ;*ADDITIONAL DEFINITIONS
                        ;*



037346   000000         WASR6:  .WORD   0               ;USED TO STORE THE STACK POINTER AFTER A TRAP
037350   000000         TRAPPC: .WORD   0               ;USED TO STORE THE PC OF A TRAP OR ABORT
037352   000000         TRAPPS: .WORD   0               ;USED TO STORE THE PS OF A TRAP OR ABORT
037354   000000         WASSR0: .WORD   0               ;USED TO STORE CONTENTS OF SR0
037356   000000         WASSR2: .WORD   0               ;USED TO STORE CONTENTS OR SR2
037360   000000         TBITPS: .WORD   0               ;SAVES THE PSW THAT MAY HAVE ITS T-BIT ON
037362   000000         $TMP0:  .WORD   0               ;TEMPORARY STORAGE LOCATION
037364   000000         $TMP1:  .WORD   0               ;TEMPORARY STORAGE LOCATION
037366   000000         $TMP2:  .WORD   0               ;TEMPORARY STORAGE LOCATION
037370   000000         $TMP3:  .WORD   0               ;TEMPORARY STORAGE LOCATION
037372   000000         $TMP4:  .WORD   0               ;TEMPORARY STORAGE LOCATION
```

```

037374  012706  001000           MMUTST: MOV     #STBOT,KSP         ;INITIALIZE THE STACK POINTER
037400  012767  021276  140642           MOV     #T0250,MMVEC       ;LOAD MEMORY MANAGENT ROUTINE INTO VECTOR
037406  012767  000340  140636           MOV     #340,MMVEC+2       ;SET NEW PS TO PRIORITY LEVEL 7-KERNEL
037414  012767  000340  177736           MOV     #340,TBITPS        ;INITIALIZE LOG THAT HOLDS T-BIT PSW
037422  005067  140144                   CLR     SR0                ;BE SURE MEM. MGMT IS OFF TO START WITH
037426  032777  000001  161670           BIT     #1,@SWR
037434  001402                           BEQ     1$
037436  000167  011060                   JMP     FPSTRT
037442  012737  050452  000030   1$:     MOV     #ERROR3,@#30       ;SET UP EMT VECTOR TO GO TO RIGHT ERROR CALL
037450  012737  000002  001004           MOV     #2,@#$TESTN        ;INCREMENT TEST NUMBER

```

```



                                ;*****************************************************************************
                                ;TEST 351         PSW PRIORITY BIT TEST
                                ;*****************************************************************************
 037456                         TS351:
 037456  005000                         CLR     R0                      ;INITIALIZE R0 WITH PRIORITY=0 DATA
 037460  005001                 2$:     CLR     R1                      ;PREPARE R1 TO ACCEPT DATA READ
 037462  106400                         MTPS    R0                      ;WRITE PRIORITY BITS IN THE PSW
 037464  106701                         MFPS    R1                      ;READ BACK THE LOW BYTE OF PSW
 037466  042701  177437                 BIC     #177437,R1              ;MASK OFF EVERYTHING EXCEPT PRIORITY BITS
 037472  020001                         CMP     R0,R1                   ;WAS CORRECT PRIORITY SET IN THE PSW?
 037474  001401                         BEQ     3$
 037476  104000                         EMT                             ;PRIORITY BITS SET WRONG IN PSW
                                                                        ;FOR TIGHTER SCOPE LOOP
                                                                        ;REPLACE ERROR CALL WITH
                                                                        ;"BR 2$" = 000770
 037500  062700  000040         3$:     ADD     #40,R0                  ;CHANGE DATA TO NEXT PRIORITY
 037504  022700  000400                 CMP     #400,R0                 ;HAVE PRIORITIES 0-7 ALL BEEN CHECKED?
 037510  001363                         BNE     2$                      ;BRANCH IF NO

                                ;*****************************************************************************
                                ;TEST 352         PSW MODE BIT TEST
                                ;*****************************************************************************
 037512                         TS352:
 037512  005000                         CLR     R0                      ;INITIALIZE R0 WITH MODE BITS = 0000
 037514  005067  140256         2$:     CLR     PSW                     ;INITIALIZE PSW
 037520  050067  140252                 BIS     R0,PSW                  ;BIT SET THE PSW MODE BITS WITH R0
 037524  016701  140246                 MOV     PSW,R1                  ;READ BACK THE CONTENTS OF THE PSW
 037530  042701  007777                 BIC     #007777,R1              ;MASK OFF EVERYTHING EXCEPT THE MODE BITS
 037534  020001                         CMP     R0,R1                   ;WERE THE MODE BITS SET CORRECTLY?
 037536  001401                         BEQ     3$
 037540  104000                         EMT                             ;MODE BITS SET WRONG IN PSW
                                                                        ;FOR TIGHTER SCOPE LOOP
                                                                        ;REPLACE ERROR CALL WITH
                                                                        ;"BR 2$" = 000763
 037542  062700  010000         3$:     ADD     #10000,R0               ;CHANGE MODE BIT DATA
 037546  001362                         BNE     2$                      ;BRANCH IF STILL MORE COMBINATIONS
 037550  005067  140222                 CLR     PSW                     ;RESET PSW BEFORE LEAVING

                                ;*****************************************************************************
                                ;TEST 353         BYTE ADDRESSING TEST FOR PSW
                                ;*****************************************************************************
 037554                         TS353:
 037554  005067  140216         2$:     CLR     PSW                     ;CLEAR THE PSW
 037560  012700  000360                 MOV     #360,R0                 ;PUT THE HIGH BYTE DATA INTO R0
 037564  110067  140207                 MOVB    R0,PSW+1                ;WRITE THE HIGH BYTE OF THE PSW
 037570  016701  140202                 MOV     PSW,R1                  ;READ BACK THE ENTIRE PSW
 037574  042701  007437                 BIC     #007437,R1              ;MASK OFF THE T & CC BITS
 037600  000300                         SWAB    R0                      ;GET DATA WRITTEN IN HIGH BYTE OF R0
 037602  020001                         CMP     R0,R1                   ;WAS THE PSW WRITTEN TO CORRECTLY
 037604  001401                         BEQ     4$
 037606  104000                         EMT                             ;LOW BYTE EFFECTED BY WRITE TO HIGH BYTE OF PSW
                                                                        ;FOR TIGHTER SCOPE LOOP
                                                                        ;REPLACE ERROR CALL WITH
                                                                        ;"BR 2$" = 000760
```

```
037610  005067  140162          4$:     CLR     PSW             ;CLEAR THE PSW
037614  012700  000340                  MOV     #340,R0         ;PUT THE LOW BYTE DATA INTO R0
037620  110067  140152                  MOVB    R0,PSW          ;WRITE THE LOW BYTE OF THE PSW
037624  016701  140146                  MOV     PSW,R1          ;READ BACK THE ENTIRE PSW
037630  042701  007437                  BIC     #007437,R1      ;MASK OFF THE T&CC BITS
037634  020001                          CMP     R0,R1           ;WAS PSW WRITTEN TO CORRECTLY
037636  001401                          BEQ     TS354
037640  104000                          EMT                     ;HIGH BYTE EFFECTED BY WRITE TO LOW BYTE OF PSW
                                                                ;FOR TIGHTER SCOPE LOOP
                                                                ;REPLACE ERROR CALL WITH
                                                                ;'BR 2$'' = 000736

                                ;******************************************************************************
                                ;TEST 354       TEST AND SETUP OF STACK POINTERS
                                ;******************************************************************************
037642                          TS354:
037642  005067  140130                  CLR     PSW             ;GO TO KERNEL MODE
037646  012706  001000                  MOV     #KERSTK,KSP     ;SET KERNEL STACK POINTER TO 1100
037652  012767  140000  140116          MOV     #140000,PSW     ;GO TO USER MODE
037660  012706  000600                  MOV     #USESTK,USP     ;SET USER STACK POINTER TO 700
037664  005067  140106                  CLR     PSW             ;BACK TO KERNEL MODE
037670  022706  001000                  CMP     #KERSTK,KSP     ;IS KERNEL R6 STILL 1100?
037674  001401                          BEQ     TS355
037676  104000                          EMT                     ;KERNEL R6 CHANGED BY WRITING USER R6
                                                                ;FOR TIGHTER SCOPE LOOP
                                                                ;REPLACE ERROR CALL WITH
                                                                ;000756

                                ;******************************************************************************
                                ;*
                                ;*      THE NEXT FIVE (5) TESTS WILL TRY TO ADDRESS ALL OF THE
                                ;*      MEMORY MANAGEMENT REGISTERS (SR0,SR1,SR2,KERNEL & USER PAR/PDR'S).
                                ;*      EVERY TIME A REGISTER TIMES OUT ITS ADDRESS WILL BE REPORTED.
                                ;*      AT THE END OF EACH TEST A SUMMARY OF THE ADDRESSES THAT TIMED
                                ;*      OUT DURING THAT TEST IS GIVEN.  THE RESULTS OF ''AND-ING'' AND ''OR-ING''
                                ;*      THEIR ADDRESSES IS GIVEN TO SHOW WHICH ADDRESS LINES MAY BE
                                ;*      STUCK AT 0 OR 1.  THE PAR/PDR ADDRESS AND KT MUX'S ARE THE
                                ;*      THINGS BEING CHECKED.
                                ;*
                                ;******************************************************************************


                                ;******************************************************************************
                                ;TEST 355       SR0,SR1,SR2,SR3 TIMEOUT TEST
                                ;******************************************************************************
037700                          TS355:
037700  012700  177572                  MOV     #SR0,R0         ;LOAD R0 WITH ADDRESS OF FIRST REG.
037704  012701  000003                  MOV     #3,R1           ;LOAD R1 WITH THE LOOP COUNT
037710  005710                  2$:     TST     (R0)            ;TRY ADDRESSING A STATUS REGISTER
                                                                ;IF IT TIMES OUT GO TO 5$
037712  062700  000002          3$:     ADD     #2,R0           ;PUT NEXT ADDRESS IN R0
037716  077104                          SOB     R1,2$           ;LOOP BACK TO 2$ UNTIL ALL TESTED
037720  005737  172516                  TST     @#172516        ;CHECK SR3 FOR RESPONSE

                                ;******************************************************************************
                                ;TEST 356       KERNEL PAR'S TIMEOUT TEST
```

```
                                ;*****************************************************************
037724                          TS356:

037724  012700  172340                  MOV     #KIPAR0,R0      ;LOAD R0 WITH ADDRESS OF FIRST REG.
037730  012701  000010                  MOV     #10,R1          ;LOAD R1 WITH LOOP COUNT (8)
037734  005710                  2$:     TST     (R0)            ;TRY ADDRESSING A KIPAR
                                                                ;IF IT TIMES OUT, WILL GO TO 5$
037736  062700  000002          3$:     ADD     #2,R0           ;PUT NEXT KIPAR ADDRESS IN R0
037742  077104                          SOB     R1,2$           ;LOOP BACK TO 2$ UNTIL ALL TESTED

                                ;*****************************************************************
                                ;TEST 357       KERNEL PDR'S TIMEOUT TEST
                                ;*****************************************************************
037744                          TS357:

037744  012700  172300                  MOV     #KIPDR0,R0      ;LOAD R0 WITH ADDRESS OF FIRST REG.
037750  012701  000010                  MOV     #10,R1          ;LOAD R1 WITH LOOP COUNT (8)
037754  005710                  2$:     TST     (R0)            ;TRY ADDRESSING A KIPDR
                                                                ;IF IT TIMES OUT, WILL GO TO 5$
037756  062700  000002          3$:     ADD     #2,R0           ;PUT NEXT KIPDR ADDRESS IN R0
037762  077104                          SOB     R1,2$           ;LOOP BACK TO 2$ UNTIL ALL TESTED

                                ;*****************************************************************
                                ;TEST 360       USER PAR'S TIMEOUT TEST
                                ;*****************************************************************
037764                          TS360:

037764  012700  177640                  MOV     #UIPAR0,R0      ;LOAD R0 WITH ADDRESS OF FIRST REG.
037770  012701  000010                  MOV     #10,R1          ;LOAD R1 WITH LOOP COUNT (8)
037774  005710                  2$:     TST     (R0)            ;TRY ADDRESSING A UIPAR
                                                                ;IF IT TIMES OUT, WILL GO TO 5$
037776  062700  000002          3$:     ADD     #2,R0           ;PUT NEXT UIPAR ADDRESS IN R0
040002  077104                          SOB     R1,2$           ;LOOP BACK TO 2$ UNTIL ALL TESTED

                                ;*****************************************************************
                                ;TEST 361       USER PDR'S TIMEOUT TEST
                                ;*****************************************************************
040004                          TS361:

040004  012700  177600                  MOV     #UIPDR0,R0      ;LOAD R0 WITH ADDRESS OF FIRST REG.
040010  012701  000010                  MOV     #10,R1          ;LOAD R1 WITH LOOP COUNT (8)
040014  005710                  2$:     TST     (R0)            ;TRY ADDRESSING A UIPDR
                                                                ;IF IT TIMES OUT, WILL GO TO 5$
040016  062700  000002          3$:     ADD     #2,R0           ;PUT NEXT UIPDR ADDRESS IN R0
040022  077104                          SOB     R1,2$           ;LOOP BACK TO 2$ UNTIL ALL TESTED

                                ;*****************************************************************
                                ;TEST 362       SR0(15:13) BIT TEST & SR2 TEST
                                ;*****************************************************************
040024                          TS362:

040024  012700  177572          1$:     MOV     #SR0,R0         ;LOAD ADDRESS OF SR0 INTO R0
040030  012710  160000                  MOV     #160000,(R0)    ;SET BITS <15:13> IN SR0 (ERROR BITS)
040034  000005                          RESET                   ;ISSUE AND "INIT" SIGNAL
040036  011001                          MOV     (R0),R1         ;READ SR0 INTO R1 TO SEE IF CLEAR
040040  001401                          BEQ     2$
```

```
040042  104000                                  EMT                             ;SR0<15:13> NOT CLEARED BY A 'RESET''
                                                                                ;FOR TIGHTER SCOPE LOOP
                                                                                ;REPLACE ERROR CALL WITH
                                                                                ;''BR 1$'' = 000770
040044  016767  137526  177304  2$:     MOV     SR2,WASSR2              ;READ CONTENTS OF SR2
040052  012701  040044                  MOV     #2$,R1                  ;LOAD EXPECTED CONTENTS INTO R1
040056  020167  177274                  CMP     R1,WASSR2               ;IS SR2 TRACKING?
040062  001401                          BEQ     3$
040064  104000                          EMT                             ;SR2 NOT ''TRACKING'' VIRTUAL ADDRESSES
                                                                        ;FOR TIGHTER SCOPE LOOP
                                                                        ;REPLACE ERROR CALL WITH
                                                                        ;''BR 2$'' = 000767
040066  012701  100000          3$:     MOV     #100000,R1              ;PUT DATA TO BE WRITTEN IN R1
040072  012703  000003                  MOV     #3,R3                   ;SETUP R3 AS A LOOP COUNTER
040076  005010                  4$:     CLR     (R0)                    ;CLEAR SR0
040100  050110                  5$:     BIS     R1,(R0)                 ;SET ONE OF THE ERROR BITS IN SR0
040102  011002                          MOV     (R0),R2                 ;READ SR0 INTO R2
040104  020102                          CMP     R1,R2                   ;DID RIGHT ERROR BIT GET SET?
040106  001401                          BEQ     6$
040110  104000                          EMT                             ;BITS WERE SET WRONG IN SR0
                                                                        ;FOR TIGHTER SCOPE LOOP
                                                                        ;REPLACE ERROR CALL WITH
                                                                        ;''BR 4$'' = 000772
040112  012704  040100          6$:     MOV     #5$,R4                  ;LOAD EXPECTED CONTENTS OF SR2 IN R4
040116  016767  137454  177232          MOV     SR2,WASSR2              ;READ SR2
040124  020467  177226                  CMP     R4,WASSR2               ;DID SR2 LOCK UP WHEN ERROR
                                                                        ;BIT SET IN SR1?
040130  001401                          BEQ     7$
040132  104000                          EMT                             ;SR2 DID NOT LOCK UP
                                                                        ;FOR TIGHTER SCOPE LOOP
                                                                        ;REPLACE ERROR CALL WITH
                                                                        ;''BR 4$'' = 000761
040134  006001                  7$:     ROR     R1                      ;CHANGE DATA TO CHECK NEXT ERROR BIT
040136  077321                          SOB     R3,4$                   ;LOOP BACK UNTIL <15:13> ALL TESTED
040140  005010                          CLR     (R0)                    ;CLEAR SR0 BEFORE LEAVING

                                ;*************************************************************************
                                ;TEST 363       SR0 & PSW DUAL ADDRESSING TEST
                                ;*************************************************************************
040142                          TS363:

040142  005067  137630          1$:     CLR     PSW                     ;CLEAR THE PSW
040146  005067  137420                  CLR     SR0                     ;CLEAR STATUS REGISTER 0
040152  106427  000340                  MTPS    #340                    ;SET PRIORITY 7 IN LOW BYTE OF PSW
040156  016700  137410                  MOV     SR0,R0                  ;READ STATUS REGISTER 0
040162  001401                          BEQ     2$
040164  104000                          EMT                             ;SR0 EFFECTED BY A WRITE TO THE PSW
                                                                        ;FOR TIGHTER SCOPE LOOP
                                                                        ;REPLACE ERROR CALL WITH
                                                                        ;''BR 1$'' = 000767
040166  005067  137400          2$:     CLR     SR0                     ;BE SURE SR0 IS 0 BEFORE LEAVING
040172  005067  137600                  CLR     PSW                     ;BE SURE PSW IS 0 BEFORE LEAVING

                                ;*************************************************************************
                                ;TEST 364       TEST THAT SR1 READS ALL ZEROS
                                ;*************************************************************************
```

```
040176                              TS364:
040176  012700  177777              1$:     MOV     #-1,R0          ;FILL R0 WITH ALL ONES
040202  016700  137366                      MOV     SR1,R0          ;READ SR1 INTO R0
040206  001401                              BEQ     2$
040210  104000                              EMT                     ;SR1 DID NOT READ ALL ZEROS
                                                                    ;FOR TIGHTER SCOPE LOOP
                                                                    ;REPLACE ERROR CALL WITH
                                                                    ;000772
040212  012767  177777  132276      2$:     MOV     #-1,SR3         ;TRY TO WRITE ONES TO SR3
040220  022767  000060  132270              CMP     #60,SR3         ;ONLY BITS <5:4> SHOULD BE ONES
040226  001401                              BEQ     3$
040230  104000                              EMT                     ;DIDN'T READ BACK A "60"
040232  004567  010170              3$:     JSR     R5,CHKAPT
040236  000402                              BR      90$
040240  000005                              RESET                   ;CLEARS SR3
040242  000402                              BR      91$
040244  005067  132246              90$:    CLR     SR3
040250  005767  132242              91$:    TST     SR3             ;VERIFY THAT IT WAS CLEARED
040254                              4$:
040254  001401                              BEQ     TS365
040256  104000                              EMT                     ;SR3 DIDN'T READ ALL ZEROS


                                    ;NOTE   F11 CHANGES INCLUDED CHECKING ALL BITS<15:0> OF PARS
                                    ;       INSTEAD OF ONLY BITS<11:0>.

                                    ;**********************************************************************
                                    ;TEST 365       BIT TEST OF KERNEL & USER PAR'S
                                    ;**********************************************************************
040260                              TS365:

040260  012700  172340              1$:     MOV     #KIPAR0,R0      ;LOAD ADDRESS OF FIRST PAR IN R0
040264  012703  000010              2$:     MOV     #10,R3          ;SETUP R3 TO COUNT 8 PAR'S
040270  005010                      3$:     CLR     (R0)            ;CLEAR THE PAR
040272  011001                              MOV     (R0),R1         ;READ THE PAR INTO R1
040274  001401                              BEQ     4$
040276  104000                              EMT                     ;PAR WOULD NOT CLEAR
                                                                    ;FOR TIGHTER SCOPE LOOP
                                                                    ;REPLACE ERROR CALL WITH
                                                                    ;"BR 3$" = 000774
040300  012704  077777              4$:     MOV     #077777,R4      ;LOAD "WALKING 0" TEST PATTERN IN R4
040304  005010                      5$:     CLR     (R0)            ;CLEAR THE PAR BEFORE LOADING DATA
040306  050410                              BIS     R4,(R0)         ;BIT SET THE TEST PATTERN INTO THE PAR
040310  011002                              MOV     (R0),R2         ;READ THE PAR INTO R2
040312  020402                              CMP     R4,R2           ;DOES DATA WRITTEN=DATA READ?
040314  001401                              BEQ     6$
040316  104000                              EMT                     ;PAR BITS DID NOT SET CORRECTLY
                                                                    ;FOR TIGHTER SCOPE LOOP
                                                                    ;REPLACE ERROR CALL WITH
                                                                    ;"BR 5$" = 000767
040320  000261                      6$:     SEC                     ;SET THE C-BIT FOR THE ROTATE INST.
040322  006004                              ROR     R4              ;ROTATE THE TEST PATTERN IN R4
040324  103767                              BCS     5$              ;BRANCH BACK IF MORE BITS TO TEST
040326  062700  000002                      ADD     #2,R0           ;GET NEXT PAR ADDRESS IN R0
040332  077322                              SOB     R3,3$           ;BRANCH BACK UNTIL ALL PAR'S TESTED
040334  022700  177660                      CMP     #UIPAR7+2,R0    ;HAVE USER PAR'S BEEN TESTED
```

```
040340  103003                  BHIS    TS366           ;GET TO NEXT TEST
040342  012700  177640          MOV     #UIPAR0,R0      ;LOAD FIRST USER PAR ADDR. IN R0
040346  000746                  BR      2$              ;BRANCH BACK TO TEST USER PAR'S
                                                        ;LEAVE TEST WITH BITS <11:1>=1 IN ALL PAR'S
                        ;*****************************************************************************
                        ;TEST 366       BIT TEST OF KERNEL & USER PDR'S
                        ;*****************************************************************************
040350                  TS366:

040350  012700  172300  1$:     MOV     #KIPDR0,R0      ;LOAD ADDRESS OF FIRST PDR IN R0
040354  012703  000010  2$:     MOV     #10,R3          ;SETUP R3 TO COUNT 8 PDR'S
040360  005010          3$:     CLR     (R0)            ;CLEAR THE PDR
040362  011001                  MOV     (R0),R1         ;READ THE PDR INTO R1
040364  001401                  BEQ     4$
040366  104000                  EMT                     ;PDR WOULD NOT CLEAR
                                                        ;FOR TIGHTER SCOPE LOOP
                                                        ;REPLACE ERROR CALL WITH
                                                        ;"BR 3$" = 000774
040370  012704  077777  4$:     MOV     #077777,R4      ;LOAD "WALKING "0" TEST PATTERN IN R4
040374  005010          5$:     CLR     (R0)            ;CLEAR THE PDR BEFORE LOADING DATA
040376  010401                  MOV     R4,R1           ;LOAD DATA INTO R1
040400  042701  100361          BIC     #100361,R1      ;MASK UNUSED BITS OUT OF THE DATA
040404  050110                  BIS     R1,(R0)         ;BIT SET THE TEST PATTERN INTO THE PDR
040406  011002                  MOV     (R0),R2         ;READ THE PDR INTO R2
040410  020102                  CMP     R1,R2           ;DOES DATA WRITTEN=DATA READ?
040412  001401                  BEQ     6$
040414  104000                  EMT                     ;PDR BITS DID NOT SET CORRECTLY
                                                        ;FOR TIGHTER SCOPE LOOP
                                                        ;REPLACE ERROR CALL WITH
                                                        ;"BR 5$" = 000767
040416  000261          6$:     SEC                     ;SET THE C-BIT FOR THE ROTATE INST.
040420  006004                  ROR     R4              ;ROTATE THE TEST PATTERN IN R4
040422  103764                  BCS     5$              ;BRANCH BACK IF MORE BITS TO TEST
040424  062700  000002          ADD     #2,R0           ;GET NEXT PDR ADDRESS IN R0
040430  077325                  SOB     R3,3$           ;BRANCH BACK UNTIL ALL PDR'S TESTED
040432  022700  177620          CMP     #UIPDR7+2,R0    ;HAVE USER PDR'S BEEN TESTED?
040436  103003                  BHIS    TS367           ;GET TO NEXT TEST
040440  012700  177600          MOV     #UIPDR0,R0      ;LOAD FIRST USER PDR ADDR. IN R0
040444  000743                  BR      2$              ;BRANCH BACK TO TEST USER PDR'S
                                                        ;LEAVE TEST WITH ALL WRITEABLE BITS IN
                                                        ;ALL PDR'S = 1


                        ;*****************************************************************************
                        ;TEST 367       TEST FOR DUAL BYTE ADDRESSING OF KERNEL & USER PAR'S
                        ;*****************************************************************************
040446                  TS367:

040446  012700  172340  1$:     MOV     #KIPAR0,R0      ;LOAD ADDRESS OF FIRST PAR INTO R0
040452  012703  000010          MOV     #10,R3          ;LOAD LOOP COUNTER TO DO 8 PAR'S
040456  012701  177777  3$:     MOV     #-1,R1          ;LOAD TEST PATTERN INTO R1
040462  005010                  CLR     (R0)            ;CLEAR THE PAR
040464  110110                  MOVB    R1,(R0)         ;WRITE 1'S TO THE LOW BYTE OF THE PAR
040466  011002                  MOV     (R0),R2         ;READ THE ENTIRE PAR INTO R2
040470  042701  177400          BIC     #177400,R1      ;MASK HIGH BYTE & UNUSED BITS OUT OF THE DATA
040474  020102                  CMP     R1,R2           ;WAS ONLY THE LOW BYTE WRITTEN TO
040476  001401                  BEQ     5$
```

```
040500  104000                          EMT                             ;HIGH BYTE EFFECTED BY WRITING LOW BYTE IN PAR
                                                                        ;FOR TIGHTER SCOPE LOOP
                                                                        ;REPLACE ERROR CALL WITH
                                                                        ;"BR 3$" = 000766
040502  005010                  5$:     CLR     (R0)                    ;CLEAR THE PAR
040504  012701  177777                  MOV     #-1,R1                  ;LOAD TEST, PATTERN INTO R1
040510  110160  000001                  MOVB    R1,1(R0)                ;WRITE 1'S TO THE HIGH BYTE OF THE PAR
040514  011002                          MOV     (R0),R2                 ;READ THE ENTIRE PAR INTO R2
                                                                        ;F11 CHANGE     WAS #170377
040516  042701  000377                  BIC     #000377,R1              ;MASK LOW BYTE & UNUSED BITS OUT OF DATA
040522  020102                          CMP     R1,R2                   ;WAS ONLY THE HIGH BYTE WRITTEN TO?
040524  001401                          BEQ     6$
040526  104000                          EMT                             ;LOW BYTE EFFECTED BY WRITING HIGH BYTE IN PAR
                                                                        ;FOR TIGHTER SCOPE LOOP
                                                                        ;REPLACE ERROR CALL WITH
                                                                        ;"BR 5" = 000765
040530  062700  000002          6$:     ADD     #2,R0                   ;PUT ADDRESS OF NEXT PAR IN R0
040534  077330                          SOB     R3,3$                   ;BRANCH BACK UNTIL 8 PAR'S TESTED
040536  022700  177660                  CMP     #UIPAR7+2,R0            ;HAVE USER PAR'S BEEN TESTED
040542  103003                          BHIS    TS370                   ;GET TO NEXT TEST
040544  012700  177640                  MOV     #UIPAR0,R0              ;LOAD ADDRESS OF FIRST USER PAR IN R0
040550  000742                          BR      3$                      ;BRANCH BACK TO TEST USER PAR'S

                                ;*****************************************************************************
                                ;TEST 370       TEST FOR DUAL BYTE ADDRESSING OF KERNEL & USER PDR'S
                                ;*****************************************************************************
040552                          TS370:

040552  012700  172300          1$:     MOV     #KIPDR0,R0              ;LOAD ADDRESS OF FIRST PDR INTO R0
040556  012703  000010                  MOV     #10,R3                  ;LOAD LOOP COUNTER TO DO 8 PDR'S
040562  012701  177777          3$:     MOV     #-1,R1                  ;LOAD TEST PATTERN INTO R1
040566  005010                          CLR     (R0)                    ;CLEAR THE PDR
040570  110110                          MOVB    R1,(R0)                 ;WRITE 1'S TO THE LOW BYTE OF THE PDR
040572  011002                          MOV     (R0),R2                 ;READ THE ENTIRE PDR INTO R2
040574  042701  177761                  BIC     #177761,R1              ;MASK HIGH BYTE & UNUSED BITS OUT OF DATA
040600  020102                          CMP     R1,R2                   ;WAS ONLY THE LOW BYTE WRITTEN TO?
040602  001401                          BEQ     5$
040604  104000                          EMT                             ;HIGH BYTE EFFECTED BY WRITING LOW BYTE IN PDR
                                                                        ;FOR TIGHTER SCOPE LOOP
                                                                        ;REPLACE ERROR CALL WITH
                                                                        ;"BR 3$" = 000766
040606  005010                  5$:     CLR     (R0)                    ;CLEAR THE PDR
040610  012701  177777                  MOV     #-1,R1                  ;LOAD TEST PATTERN INTO R1
040614  110160  000001                  MOVB    R1,1(R0)                ;WRITE 1'S TO THE HIGH BYTE OF THE PDR
040620  011002                          MOV     (R0),R2                 ;READ THE ENTIRE PDR INTO R2
040622  042701  100377                  BIC     #100377,R1              ;MASK LOW BYTE & UNUSED BITS OUT OF DATA
040626  020102                          CMP     R1,R2                   ;WAS ONLY THE HIGH BYTE WRITTEN TO?
040630  001401                          BEQ     6$
040632  104000                          EMT                             ;LOW BYTE EFFECTED BY WRITING HIGH BYTE IN PDR
                                                                        ;FOR TIGHTER SCOPE LOOP
                                                                        ;REPLACE ERROR CALL WITH
                                                                        ;"BR 5$" = 000765
040634  062700  000002          6$:     ADD     #2,R0                   ;PUT ADDRESS OF NEXT PDR IN R0
040640  077330                          SOB     R3,3$                   ;BRANCH BACK UNTIL 8 PDR'S TESTED
040642  022700  177620                  CMP     #UIPDR7+2,R0            ;HAVE USER PDR'S BEEN TESTED?
040646  103003                          BHIS    TS371                   ;GET TO NEXT TEST
```

```
040650  012700  177600                 MOV     #UIPDR0,R0      ;LOAD ADDRESS OF FIRST USER PDR IN R0
040654  000742                         BR      3$              ;BRANCH BACK TO TEST USER PDR'S

                               ;*************************************************************************
                               ;TEST 371       PAR-PDR DUAL ADDRESSING TEST
                               ;*************************************************************************
040656                         TS371:

040656  012703  000010                 MOV     #10,R3          ;LOAD LOOP COUNTER WITH AN 8
040662  012700  172300                 MOV     #KIPDR0,R0      ;LOAD ADDRESS OF FIRST KERNEL PDR AND R0
040666  004767  007260                 JSR     PC,SETREG       ;SET ALL BITS IN ALL PAR'S IN PDR'S
040672  012706  001000         2$:     MOV     #KERSTK,KSP     ;SETUP STACK POINTER
040676  005010                         CLR     (R0)            ;CLEAR ONE OF THE KERNEL PDR'S
040700  004767  007340                 JSR     PC,CMPREG       ;SEE IF OTHER PAR/PDR'S WERE EFFECTED
040704  012720  177777                 MOV     #-1,(R0)+       ;RESTORE ALL ONES, AND SETUP FOR NEXT PDR
040710  077310                         SOB     R3,2$           ;LOOP TO 2$ UNTIL ALL KERNEL PDR'S CHECKED
040712  012703  000010                 MOV     #10,R3          ;LOAD LOOP COUNTER WITH AN 8
040716  012700  172340                 MOV     #KIPAR0,R0      ;LOAD ADDRESS OF FIRST KERNEL PAR IN R0
040722  012706  001000         3$:     MOV     #KERSTK,KSP     ;SETUP STACK POINTER
040726  005010                         CLR     (R0)            ;CLEAR ONE OF THE KERNEL PAR'S
040730  004767  007310                 JSR     PC,CMPREG       ;SEE IF OTHER PAR/PDR'S WERE EFFECTED
040734  012720  177777                 MOV     #-1,(R0)+       ;RESTORE ALL ONES, AND SETUP FOR NEXT PAR
040740  077310                         SOB     R3,3$           ;LOOP TO 3$ UNTIL ALL KERNEL PAR'S CHECKED
040742  012703  000010                 MOV     #10,R3          ;LOAD LOOP COUNTER WITH AN 8
040746  012700  177600                 MOV     #UIPDR0,R0      ;LOAD ADDRESS OF FIRST USER PDR IN R0
040752  012706  001000         4$:     MOV     #KERSTK,KSP     ;SETUP STACK POINTER
040756  005010                         CLR     (R0)            ;CLEAR ONE OF THE USER PDR'S
040760  004767  007260                 JSR     PC,CMPREG       ;SEE IF OTHER PAR/PDR'S WERE EFFECTED
040764  012720  177777                 MOV     #-1,(R0)+       ;RESTORE ALL ONES, AND SETUP FOR NEXT UPDR
040770  077310                         SOB     R3,4$           ;LOOP TO 4$ UNTIL ALL USER PDR'S CHECKED
040772  012703  000010                 MOV     #10,R3          ;LOAD LOOP COUNTER WITH AN 8
040776  012700  177640                 MOV     #UIPAR0,R0      ;LOAD ADDRESS OF FIRST USER PAR IN R0
041002  012706  001000         5$:     MOV     #KERSTK,KSP     ;SETUP STACK POINTER
041006  005010                         CLR     (R0)            ;CLEAR ONE OF THE USER PAR'S
041010  004767  007230                 JSR     PC,CMPREG       ;SEE IF OTHER PAR/PDR'S WERE EFFECTED
041014  012720  177777                 MOV     #-1,(R0)+       ;RESTORE ALL ONES, AND SETUP FOR NEXT UPAR
041020  077310                         SOB     R3,5$           ;LOOP TO 5$ UNTIL ALL USER PAR'S CHECKED

                               ;*************************************************************************
                               ;TEST 372       TEST THAT PAR-PDR'S NOT AFFECTED BY RESET
                               ;*************************************************************************
041022                         TS372:


041022  032737  000001  001020         BIT     #1, @#$ENV      ;ARE WE RUNNING UNDER API
041030  001403                         BEQ     70$             ;IF NO THEN DO TEST
041032  005737  001006                 TST     @#$PASS         ;IS THIS FIRST PASS
041036  001063                         BNE     TS373           ;IF NO THEN SHIP TO NEXT TEST
041040                         70$:
041040  004767  007106         1$:     JSR     PC,SETREG       ;SET ALL BITS IN ALL PAR'S AND PDR'S
041044  000005                         RESET                   ;ISSUE AN ''INIT'' BY EXECUTING A RESET
041046  012700  172300         10$:    MOV     #KIPDR0,R0      ;LOAD ADDRESS OF FIRST KERNEL PDR IN R0
041052  012704  000010                 MOV     #10,R4          ;LOAD LOOP COUNTER WITH AN 8
041056  011001                 2$:     MOV     (R0),R1         ;READ A KERNEL PDR INTO R1
041060  022701  077416                 CMP     #77416,R1       ;ARE ALL THE BITS STILL SET?
041064  001401                         BEQ     3$
```

```
041066  104000                  EMT                     ;KERNEL PDR AFFECTED BY A RESET
                                                        ;FOR TIGHTER SCOPE LOOP
                                                        ;REPLACE ERROR CALL WITH
                                                        ;"BR 2$" = 000773
041070  062700  000002  3$:     ADD     #2,R0           ;FORM ADDRESS OF NEXT KERNEL PDR
041074  077410                  SOB     R4,2$           ;LOOP TO 2$ UNTIL ALL KERNEL PDR'S CHECKED
041076  012700  172340          MOV     #KIPAR0,R0      ;LOAD ADDRESS OF FIRST KERNEL PAR IN R0
041102  012704  000010          MOV     #10,R4          ;LOAD LOOP COUNTER WITH AN 8
041106  011001          4$:     MOV     (R0),R1         ;READ A KERNEL PAR INTO R1
                                                        ;****F11 CHANGE****     WAS #7777
041110  022701  177777          CMP     #177777,R1      ;ARE ALL THE BITS STILL SET?
041114  001401                  BEQ     5$
041116  104000                  EMT                     ;KERNEL PAR AFFECTED BY A RESET
                                                        ;FOR TIGHTER SCOPE LOOP
                                                        ;REPLACE ERROR CALL WITH
                                                        ;"BR 4$" = 000773
041120  062700  000002  5$:     ADD     #2,R0           ;FORM ADDRESS OF NEXT KERNEL PAR
041124  077410                  SOB     R4,4$           ;LOOP TO 4$ UNTIL ALL KERNEL PAR'S CHECKED
041126  012700  177600          MOV     #UIPDR0,R0      ;LOAD ADDRESS OF FIRST USER PDR IN R0
041132  012704  000010          MOV     #10,R4          ;LOAD LOOP COUNTER WITH AN 8
041136  011001          6$:     MOV     (R0),R1         ;READ A USER PDR INTO R1
041140  022701  077416          CMP     #77416,R1       ;ARE ALL THE BITS STILL SET?
041144  001401                  BEQ     7$
041146  104000                  EMT                     ;USER PDR AFFECTED BY A RESET
                                                        ;FOR TIGHTER SCOPE LOOP
                                                        ;REPLACE ERROR CALL WITH
                                                        ;"BR 6$" = 000773
041150  062700  000002  7$:     ADD     #2,R0           ;FORM ADDRESS OF NEXT USER PDR
041154  077410                  SOB     R4,6$           ;LOOP TO 6$ UNTIL ALL USER PDR'S CHECKED

041156  012700  177640          MOV     #UIPAR0,R0      ;LOAD ADDRESS OF FIRST USER PAR IN R0
041162  012704  000010          MOV     #10,R4          ;LOAD LOOP COUNTER WITH AN 8
041166  011001          8$:     MOV     (R0),R1         ;READ A USER PAR INTO R1
                                                        ;****F11 CHANGE****     WAS #7777
041170  022701  177777          CMP     #177777,R1      ;ARE ALL THE BITS STILL SET?
041174  001401                  BEQ     9$
041176  104000                  EMT                     ;USER PAR AFFECTED BY A RESET
                                                        ;FOR TIGHTER SCOPE LOOP
                                                        ;REPLACE ERROR CALL WITH
                                                        ;"BR 8$" = 000773
041200  062700  000002  9$:     ADD     #2,R0           ;FORM ADDRESS OF NEXT USER PAR
041204  077410                  SOB     R4,8$           ;LOOP TO 8$ UNTIL ALL USER PAR'S CHECKED
```

```
                                        ;*******************************************************************************
                                        ;TEST 373       RELOCATION & ADDER TEST (NO CARRIES)
                                        ;*******************************************************************************
041206                                  TS373:

041206  012700  172340                  1$:     MOV     #KIPAR0,R0      ;LOAD ADDRESS OF FIRST KERNEL PAR IN R0
041212  005001                                  CLR     R1              ;CLEAR R1
041214  012702  000007                          MOV     #7,R2           ;LOAD LOOP COUNTER WITH A 7
041220  010120                          2$:     MOV     R1,(R0)+        ;MAP KERNEL PAR'S TO PAGES 0-6 (4K EACH)
041222  062701  000200                          ADD     #200,R1
041226  077204                                  SOB     R2,2$           ;LOOP UNTIL KIPAR0 - KIPAR6 ARE LOADED
041230  012710  177600                          MOV     #177600,(R0)    ;MAP KIPAR7 TO THE I/O PAGE
041234  012700  172300                          MOV     #KIPDR0,R0      ;LOAD ADDRESS OF FIRST KERNEL PDR IN R0
041240  012701  077406                          MOV     #77406,R1       ;LOAD PDR DATA INTO R1
041244  012702  000010                          MOV     #10,R2          ;LOAD LOOP COUNTER WITH AN 8
041250  010120                          3$:     MOV     R1,(R0)+        ;MAP ALL 8 PAGES 128 BLOCKS, UPWARD
041252  077202                                  SOB     R2,3$           ;       EXPANDABLE, READ/WRITE

041254  012700  067776                  4$:     MOV     #67776,R0       ;LOAD PHYSICAL ADDR. PBA INTO R0
041260  012701  107776                          MOV     #107776,R1      ;LOAD VIRTUAL ADDR. VBA INTO R1
041264  012702  125250                          MOV     #125250,R2      ;LOAD TEST PATTERN INTO R2
041270  012704  000600                          MOV     #600,R4 ;LOAD R4 WITH PAR VALUE
041274  010467  131050                          MOV     R4,KIPAR4       ;LOAD KERNEL PAR 4 BITS <11:00>
041300  011067  176056                          MOV     (R0),$TMP0      ;SAVE CONTENTS AT TEST LOCATION
041304  005067  131206                          CLR     SR3             ;SET UP FOR 18-BIT ADDRESSING
041310  052767  000001  136254                  BIS     #BIT0,SR0       ;TURN ON "RELOCATION"
041316  010211                                  MOV     R2,(R1)         ;LOAD 125250 USING ADDER  (PAR4 + VIRT ADDR.)
041320  005067  136246                          CLR     SR0             ;TURN OFF MEMORY MGMT.
041324  011003                                  MOV     (R0),R3         ;READ 125250 BACK WITHOUT USING MEM. MGMT.
041326  016710  176030                          MOV     $TMP0,(R0)      ;RESTORE ORIGINAL CONTENTS TO TEST LOC.
041332  020203                                  CMP     R2,R3           ;WAS SAME PATTERN READ BACK THAT WAS
                                                                        ;WRITTEN USING "DEST-ONLY-RELOC."?
041334  001401                                  BEQ     5$
041336  104000                                  EMT                     ;TEST LOCATION DID NOT HAVE PATTERN
                                                                        ;THAT SHOULD HAVE BEEN WRITTEN TO IT.
                                                                        ;APPARENTLY PHYSICAL ADDR. WAS
                                                                        ;FORMED WRONG BY ADDERS USING
                                                                        ;THE VIRTUAL ADDR. AND KIPAR4
                                                                        ;FOR TIGHTER SCOPE LOOP
                                                                        ;REPLACE ERROR CALL WITH
                                                                        ;"BR 4$" = 000742
041340                                  5$:
041340  012700  067776                  6$:     MOV     #67776,R0       ;LOAD PHYSICAL ADDR. PBA INTO R0
041344  012701  102576                          MOV     #102576,R1      ;LOAD VIRTUAL ADDR. VBA INTO R1
041350  012702  125251                          MOV     #125251,R2      ;LOAD TEST PATTERN INTO R2
041354  012704  000652                          MOV     #652,R4 ;LOAD R4 WITH PAR VALUE
041360  010467  130764                          MOV     R4,KIPAR4       ;LOAD KERNEL PAR 4 BITS <11:00>
```

```
041364 011067 175772                MOV     (R0),$TMP0      ;SAVE CONTENTS AT TEST LOCATION
041370 005067 131122                CLR     SR3             ;SET UP FOR 18-BIT ADDRESSING
041374 052767 000001 136170         BIS     #BIT0,SR0       ;TURN ON "RELOCATION"
041402 010211                       MOV     R2,(R1)         ;LOAD 125251 USING ADDER  (PAR4 + VIRT ADDR.)
041404 005067 136162                CLR     SR0             ;TURN OFF MEMORY MGMT.
041410 011003                       MOV     (R0),R3         ;READ 125251 BACK WITHOUT USING MEM. MGMT.
041412 016710 175744                MOV     $TMP0,(R0)      ;RESTORE ORIGINAL CONTENTS TO TEST LOC.
041416 020203                       CMP     R2,R3           ;WAS SAME PATTERN READ BACK THAT WAS
                                                            ;WRITTEN USING 'DEST-ONLY-RELOC.'?
041420 001401                       BEQ     7$
041422 104000                       EMT                     ;TEST LOCATION DID NOT HAVE PATTERN
                                                            ;THAT SHOULD HAVE BEEN WRITTEN TO IT.
                                                            ;APPARENTLY PHYSICAL ADDR. WAS
                                                            ;FORMED WRONG BY ADDERS USING
                                                            ;THE VIRTUAL ADDR. AND KIPAR4
                                                            ;FOR TIGHTER SCOPE LOOP
                                                            ;REPLACE ERROR CALL WITH
                                                            ;"BR 6$" = 000742
041424                        7$:
041424 012700 067776          8$:   MOV     #67776,R0       ;LOAD PHYSICAL ADDR. PBA INTO R0
041430 012701 105276                MOV     #105276,R1      ;LOAD VIRTUAL ADDR. VBA INTO R1
041434 012702 125252                MOV     #125252,R2      ;LOAD TEST PATTERN INTO R2
041440 012704 000625                MOV     #625,R4 ;LOAD R4 WITH PAR VALUE
041444 010467 130700                MOV     R4,KIPAR4       ;LOAD KERNEL PAR 4 BITS <11:00>
041450 011067 175706                MOV     (R0),$TMP0      ;SAVE CONTENTS AT TEST LOCATION
041454 052767 000020 131034         BIS     #BIT4,SR3       ;SET UP FOR 22-BIT ADDRESSING
041462 052767 000001 136102         BIS     #BIT0,SR0       ;TURN ON "RELOCATION"
041470 010211                       MOV     R2,(R1)         ;LOAD 125252 USING ADDER  (PAR4 + VIRT ADDR.)
041472 005067 136074                CLR     SR0             ;TURN OFF MEMORY MGMT.
041476 011003                       MOV     (R0),R3         ;READ 125252 BACK WITHOUT USING MEM. MGMT.
041500 016710 175656                MOV     $TMP0,(R0)      ;RESTORE ORIGINAL CONTENTS TO TEST LOC.
041504 020203                       CMP     R2,R3           ;WAS SAME PATTERN READ BACK THAT WAS
                                                            ;WRITTEN USING 'DEST-ONLY-RELOC.'?
041506 001401                       BEQ     9$
041510 104000                       EMT                     ;TEST LOCATION DID NOT HAVE PATTERN
                                                            ;THAT SHOULD HAVE BEEN WRITTEN TO IT.
                                                            ;APPARENTLY PHYSICAL ADDR. WAS
                                                            ;FORMED WRONG BY ADDERS USING
                                                            ;THE VIRTUAL ADDR. AND KIPAR4
                                                            ;FOR TIGHTER SCOPE LOOP
                                                            ;REPLACE ERROR CALL WITH
                                                            ;"BR 8$" = 000742
041512                        9$:

041512 012700 177776                MOV     #PSW,R0         ;LOAD PHYS. ADDR. OF PSW INTO R0
041516 012701 100076                MOV     #100076,R1      ;LOAD VIRTUAL ADDR. FOR PSW INTO R1
041522 012702 030340                MOV     #030340,R2      ;LOAD DATA FOR PSW IN R2
041526 012704 007777                MOV     #7777,R4        ;LOAD R4 WITH PAR VALUE
041532 010467 130612                MOV     R4,KIPAR4       ;LOAD KERNEL PAR 4 BITS <11:00>
041536 005010                       CLR     (R0)            ;CLEAR THE PSW
041540 005067 130752                CLR     SR3             ;SET UP FOR 18-BIT ADDRESSING
041544 052767 000001 136020         BIS     #BIT0,SR0       ;TURN ON "MEMORY MANAGEMENT"
041552 010211                       MOV     R2,(R1)         ;LOAD PSW USING ADDER (PAR4 + VIRT ADDR.)
041554 005067 136012                CLR     SR0             ;TURN OFF MEM. MGMT (SR0=0)
041560 011003                       MOV     (R0),R3         ;READ PSW BACK WITHOUT USING MEM. MGMT.
041562 005010                       CLR     (R0)            ;CLEAR THE PSW
```

```
041564  042703  000037                  BIC     #37,R3          ;MASK T-BIT & CC BITS OUT OF DATA READ
041570  020203                          CMP     R2,R3           ;WAS PSW WRITTEN?
041572  001401                          BEQ     11$
041574  104000                          EMT                     ;PSW DID NOT HAVE DATA THAT IT SHOULD HAVE,
                                                                ;APPARENTLY PHYS. ADDR. OF PSW WAS
                                                                ;NOT FORMED BY ADDERS USING THE
                                                                ;VIRTUAL ADDR. AND KIPAR4
                                                                ;FOR TIGHTER SCOPE LOOP
                                                                ;REPLACE ERROR CALL WITH
                                                                ;"BR 10$" = 000742
041576  012700  177776          11$:    MOV     #PSW,R0         ;LOAD PHYS. ADDR. OF PSW INTO R0
041602  012701  117776                  MOV     #117776,R1      ;LOAD VIRTUAL ADDR. FOR PSW INTO R1
041606  012702  030240                  MOV     #030240,R2      ;LOAD DATA FOR PSW IN R2
041612  012704  177600                  MOV     #177600,R4      ;LOAD R4 WITH PAR VALUE
041616  010467  130526                  MOV     R4,KIPAR4       ;LOAD KERNEL PAR 4 BITS <11:00>
041622  052767  000020  130666          BIS     #BIT4,SR3       ;SET UP FOR 22-BIT ADDRESSING
041630  052767  000001  135734          BIS     #BIT0,SR0       ;TURN ON "MEMORY MANAGEMENT"
041636  010211                          MOV     R2,(R1)         ;LOAD PSW USING ADDEP (PAR4 + VIRT. ADDR.)
041640  005067  135726                  CLR     SR0             ;TURN OFF MEM. MGMT (SR0=0)
041644  011003                          MOV     (R0),R3         ;READ PSW BACK WITHOUT USING MEM. MGMT.
041646  005010                          CLR     (R0)            ;CLEAR THE PSW
041650  042703  000037                  BIC     #37,R3          ;MASK T-BIT & CC BITS OUT OF DATA READ
041654  020203                          CMP     R2,R3           ;WAS PSW WRITTEN WHILE IN MAINT. MODE?
041656  001401                          BEQ     TS374
041660  104000                          EMT                     ;PSW DID NOT HAVE DATA THAT IT SHOULD
                                                                ;HAVE, APPARENTLY PHYS. ADDR. OF PSW WAS
                                                                ;NOT FORMED BY ADDERS USING THE
                                                                ;VIRTUAL ADDR. AND KIPAR4
                                                                ;FOR TIGHTER SCOPE LOOP
                                                                ;REPLACE ERROR CALL WITH
                                                                ;"BR 11$" = 000743

                                ;*****************************************************************************
                                ;TEST 374       RELOCATION & ADDER TEST (WITH CARRIES)
                                ;*****************************************************************************
041662                          TS374:

041662                          1$:                             ;KERNEL PAR'S AND PDR'S HAVE BEEN
                                                                ;SETUP BY THE PREVIOUS TEST
041662  012700  066476          2$:     MOV     #66476,R0       ;LOAD PHYSICAL ADDR. PBA INTO R0
041666  012701  114376                  MOV     #114376,R1      ;LOAD VIRTUAL ADDR. VBA INTO R1
041672  012702  125253                  MOV     #125253,R2      ;LOAD TEST PATTERN INTO R2
041676  012704  000521                  MOV     #521,R4 ;LOAD R4 WITH PAR VALUE
041702  010467  130442                  MOV     R4,KIPAR4       ;LOAD KERNEL PAR 4 BITS <11:00>
041706  011067  175450                  MOV     (R0),$TMP0      ;SAVE CONTENTS AT TEST LOCATION
041712  052767  000020  130576          BIS     #BIT4,SR3       ;SET UP FOR 22-BIT ADDRESSING
041720  052767  000001  135644          BIS     #BIT0,SR0       ;TURN ON "RELOCATION"
041726  010211                          MOV     R2,(R1)         ;LOAD 125253 USING ADDER  (PAR4 + VIRT ADDR.)
041730  005067  135636                  CLR     SR0             ;TURN OFF MEMORY MGMT.
041734  011003                          MOV     (R0),R3         ;READ 125253 BACK WITHOUT USING MEM. MGMT.
041736  016710  175420                  MOV     $TMP0,(R0)      ;RESTORE ORIGINAL CONTENTS TO TEST LOC.
041742  020203                          CMP     R2,R3           ;WAS SAME PATTERN READ BACK THAT WAS
                                                                ;WRITTEN USING "DEST-ONLY-RELOC."?
041744  001401                          BEQ     3$
041746  104000                          EMT                     ;TEST LOCATION DID NOT HAVE PATTERN
                                                                ;THAT SHOULD HAVE BEEN WRITTEN TO IT.
```

```
                                                                ;APPARENTLY PHYSICAL ADDR. WAS
                                                                ;FORMED WRONG BY ADDERS USING
                                                                ;THE VIRTUAL ADDR. AND KIPAR4
                                                                ;FOR TIGHTER SCOPE LOOP
                                                                ;REPLACE ERROR CALL WITH
                                                                ;'BR 2$' = 000742
041750                                  3$:
041750  012700  062276                  4$:     MOV     #62276,R0       ;LOAD PHYSICAL ADDR. PBA INTO R0
041754  012701  107376                          MOV     #107376,R1      ;LOAD VIRTUAL ADDR. VBA INTO R1
041760  012702  125254                          MOV     #125254,R2      ;LOAD TEST PATTERN INTO R2
041764  012704  000527                          MOV     #527,R4 ;LOAD R4 WITH PAR VALUE
041770  010467  130354                          MOV     R4,KIPAR4       ;LOAD KERNEL PAR 4 BITS <11:00>
041774  011067  175362                          MOV     (R0),$TMP0      ;SAVE CONTENTS AT TEST LOCATION
042000  052767  000020  130510                  BIS     #BIT4,SR3       ;SET UP FOR 22-BIT ADDRESSING
042006  052767  000001  135556                  BIS     #BIT0,SR0       ;TURN ON "RELOCATION"
042014  010211                                  MOV     R2,(R1)         ;LOAD 125254 USING ADDER  (PAR4 + VIRT ADDR.)
042016  005067  135550                          CLR     SR0             ;TURN OFF MEMORY MGMT.
042022  011003                                  MOV     (R0),R3         ;READ 125254 BACK WITHOUT USING MEM. MGMT.
042024  016710  175332                          MOV     $TMP0,(R0)      ;RESTORE ORIGINAL CONTENTS TO TEST LOC.
042030  020203                                  CMP     R2,R3           ;WAS SAME PATTERN READ BACK THAT WAS
                                                                        ;WRITTEN USING 'DEST-ONLY-RELOC.'?
042032  001401                                  BEQ     5$
042034  104000                                  EMT
                                                                        ;TEST LOCATION DID NOT HAVE PATTERN
                                                                        ;THAT SHOULD HAVE BEEN WRITTEN TO IT.
                                                                        ;APPARENTLY PHYSICAL ADDR. WAS
                                                                        ;FORMED WRONG BY ADDERS USING
                                                                        ;THE VIRTUAL ADDR. AND KIPAR4
                                                                        ;FOR TIGHTER SCOPE LOOP
                                                                        ;REPLACE ERROR CALL WITH
                                                                        ;'BR 4$' = 000742
042036                                  5$:
042036  012700  062076                  6$:     MOV     #62076,R0       ;LOAD PHYSICAL ADDR. PBA INTO R0
042042  012701  104576                          MOV     #104576,R1      ;LOAD VIRTUAL ADDR. VBA INTO R1
042046  012702  125255                          MOV     #125255,R2      ;LOAD TEST PATTERN INTO R2
042052  012704  000553                          MOV     #553,R4 ;LOAD R4 WITH PAR VALUE
042056  010467  130266                          MOV     R4,KIPAR4       ;LOAD KERNEL PAR 4 BITS <11:00>
042062  011067  175274                          MOV     (R0),$TMP0      ;SAVE CONTENTS AT TEST LOCATION
042066  052767  000020  130422                  BIS     #BIT4,SR3       ;SET UP FOR 22-BIT ADDRESSING
042074  052767  000001  135470                  BIS     #BIT0,SR0       ;TURN ON "RELOCATION"
042102  010211                                  MOV     R2,(R1)         ;LOAD 125255 USING ADDER  (PAR4 + VIRT ADDR.)
042104  005067  135462                          CLR     SR0             ;TURN OFF MEMORY MGMT.
042110  011003                                  MOV     (R0),R3         ;READ 125255 BACK WITHOUT USING MEM. MGMT.
042112  016710  175244                          MOV     $TMP0,(R0)      ;RESTORE ORIGINAL CONTENTS TO TEST LOC.
042116  020203                                  CMP     R2,R3           ;WAS SAME PATTERN READ BACK THAT WAS
                                                                        ;WRITTEN USING 'DEST-ONLY-RELOC.'?
042120  001401                                  BEQ     7$
042122  104000                                  EMT
                                                                        ;TEST LOCATION DID NOT HAVE PATTERN
                                                                        ;THAT SHOULD HAVE BEEN WRITTEN TO IT.
                                                                        ;APPARENTLY PHYSICAL ADDR. WAS
                                                                        ;FORMED WRONG BY ADDERS USING
                                                                        ;THE VIRTUAL ADDR. AND KIPAR4
                                                                        ;FOR TIGHTER SCOPE LOOP
                                                                        ;REPLACE ERROR CALL WITH
                                                                        ;'BR 6$' = 000742
042124                                  7$:
042124  012700  000000                  8$:     MOV     #000000,R0      ;LOAD PHYSICAL ADDR. PBA INTO R0
```

```
042130  012701  111400                      MOV     #111400,R1       ;LOAD VIRTUAL ADDR. VBA INTO R1
042134  012702  125256                      MOV     #125256,R2       ;LOAD TEST PATTERN INTO R2
042140  012704  177664                      MOV     #177664,R4       ;LOAD R4 WITH PAR VALUE
042144  010467  130200                      MOV     R4,KIPAR4        ;LOAD KERNEL PAR 4 BITS <11:00>
042150  011067  175206                      MOV     (R0),$TMPO       ;SAVE CONTENTS AT TEST LOCATION
042154  052767  000020  130334              BIS     #BIT4,SR3        ;SET UP FOR 22-BIT ADDRESSING
042162  052767  000001  135402              BIS     #BIT0,SR0        ;TURN ON "RELOCATION"
042170  010211                              MOV     R2,(R1)          ;LOAD 125256 USING ADDER  (PAR4 + VIRT ADDR.)
042172  005067  135374                      CLR     SR0              ;TURN OFF MEMORY MGMT.
042176  011003                              MOV     (R0),R3          ;READ 125256 BACK WITHOUT USING MEM. MGMT.
042200  016710  175156                      MOV     $TMPO,(R0)       ;RESTORE ORIGINAL CONTENTS TO TEST LOC.
042204  020203                              CMP     R2,R3            ;WAS SAME PATTERN READ BACK THAT WAS
                                                                     ;WRITTEN USING "DEST-ONLY-RELOC."?
042206  001401                              BEQ     9$
042210  104000                              EMT                      ;TEST LOCATION DID NOT HAVE PATTERN
                                                                     ;THAT SHOULD HAVE BEEN WRITTEN TO IT.
                                                                     ;APPARENTLY PHYSICAL ADDR. WAS
                                                                     ;FORMED WRONG BY ADDERS USING
                                                                     ;THE VIRTUAL ADDR. AND KIPAR4
                                                                     ;FOR TIGHTER SCOPE LOOP
                                                                     ;REPLACE ERROR CALL WITH
                                                                     ;"BR 8$" = 000742
042212                              9$:

                                    ;*****************************************************************************
                                    ;TEST 375       READ AND WRITE WHILE IN RELOCATE MODE
                                    ;*****************************************************************************
042212                              TS375:

042212  005067  135560              1$:     CLR     PSW              ;START IN KERNEL MODE
042216  012704  001377                      MOV     #1377,R4         ;LOAD R4 WITH VALUE FOR PAR4
042222  012705  001400                      MOV     #1400,R5         ;LOAD R5 WITH VALUE FOR PAR5
042226  010467  130116                      MOV     R4,KIPAR4        ;LOAD KERNEL PAR4
042232  010567  130114                      MOV     R5,KIPAR5        ;LOAD KERNEL PAR5
042236  012700  177640                      MOV     #UIPAR0,R0       ;LOAD ADDRESS OF FIRST USER PAR IN R0
042242  005001                              CLR     R1               ;CLEAR R1
042244  012702  000007                      MOV     #7,R2            ;LOAD LOOP COUNTER WITH A 7
042250  010120                      2$:     MOV     R1,(R0)+         ;MAP USER PAR'S TO PAGES 0-6 (4K EACH)
042252  062701  000200                      ADD     #200,R1
042256  077204                              SOB     R2,2$            ;LOOP UNTIL UIPAR0-UIPAR6 ARE LOADED
042260  012710  177600                      MOV     #177600,(R0)     ;MAP USER PAR7 TO THE I/O PAGE
042264  012700  177600                      MOV     #UIPDR0,R0       ;LOAD ADDRESS OF FIRST USER PDR IN R0
042270  012701  077406                      MOV     #77406,R1        ;LOAD PDR DATA INTO R1
042274  012702  000010                      MOV     #10,R2           ;LOAD LOOP COUNTER WITH AN 8
042300  010120                      3$:     MOV     R1,(R0)+         ;MAP ALL 8 PAGES 128 BLOCKS, UPWARD
042302  077202                              SOB     R2,3$            ;      EXPANDABLE, READ/WRITE
042304  012767  042550  135736              MOV     #8$,MMVEC        ;SET M. M. TRAP VECTOR TO 8$
042312  052767  000020  130176              BIS     #BIT4,SR3        ;SET UP FOR 22-BIT ADDRESSING
042320  012767  000001  135244              MOV     #BIT0,SR0        ;TURN ON MEMORY MANAGEMENT
042326  105067  135256                      CLRB    UIPDR4           ;MAP USER SPACE NON-RESIDENT WHILE
042332  105067  135254                      CLRB    UIPDR5           ;      TESTING KERNEL SPACE
042336  010567  135306                      MOV     R5,UIPAR4        ;MAP USER PAR'S OPPOSITE OF KIPAR'S
042342  010467  135304                      MOV     R4,UIPAR5
042346  016767  135424  175006      4$:     MOV     PSW,$TMPO        ;SAVE PSW IN CASE OF ERROR
042354  012700  100100                      MOV     #100100,R0       ;PUT VIRTUAL ADDR. THAT USES PAR4 IN R0
042360  012701  120000                      MOV     #120000,R1       ;PUT VIRTUAL ADDR. THAT USES PAR5 IN R1
```

```
042364  010010                  5$:     MOV     R0,(R0)         ;WRITE TO TEST LOC. USING PAR4
042366  011102                          MOV     (R1),R2         ;READ THE SAME LOC., BUT USING PAR5
042370  020002                          CMP     R0,R2           ;DID WE READ WHAT WE WROTE?
042372  001401                          BEQ     6$
042374  104000                          EMT                     ;READING LOC. USING PAR5 AND A VIRT.
                                                                ;ADDR. DID NOT FIND DATA WRITTEN WHEN USING
                                                                ;PAR4 AND VIRT. ADDRESS.
                                                                ;FOR TIGHTER SCOPE LOOP
                                                                ;REPLACE ERROR CALL WITH
                                                                ;"BR 5$" = 000765
042376  062700  000100          6$:     ADD     #100,R0         ;CHANGE VIRTUAL ADDRS. TO POINT TO NEXT BLOCK
042402  062701  000100                  ADD     #100,R1
042406  020127  127700                  CMP     R1,#127700      ;WERE BLOCKS FROM 60000-676000 ALL TRIED?
042412  001364                          BNE     5$              ;BRANCH IF NO
042414  032767  140000  135354          BIT     #140000,PSW     ;HAVE WE DONE TEST IN USER MODE YET?
042422  001026                          BNE     7$              ;BRANCH IF YES
042424  010467  135220                  MOV     R4,UIPAR4       ;LOAD USER PAR4
042430  010567  135216                  MOV     R5,UIPAR5       ;LOAD USER PAR5
042434  112767  000006  135146          MOVB    #6,UIPDR4       ;MAP USER SPACE R/W TO TEST IT
042442  112767  000006  135142          MOVB    #6,UIPDR5
042450  105067  127634                  CLRB    KIPDR4          ;MAP  KERNEL SPACE NON-RESIDENT WHILE
042454  105067  127632                  CLRB    KIPDR5          ;       TESTING USER SPACE
042460  010567  127664                  MOV     R5,KIPAR4       ;MAP KERNEL PAR'S OPPOSITE UIPAR'S
042464  010467  127662                  MOV     R4,KIPAR5
042470  012767  140000  135300          MOV     #140000,PSW     ;GO TO USER MODE
042476  000723                          BR      4$              ;GO BACK AND READ/WRITE IN USER MODE
042500  005067  135272          7$:     CLR     PSW             ;GO BACK TO KERNEL MODE BEFORE LEAVING
042504  012767  077406  127576          MOV     #77406,KIPDR4   ;REMAP KERNEL PAGES READ/WRITE
042512  012767  077406  127572          MOV     #77406,KIPDR5
042520  010567  127624                  MOV     R5,KIPAR4       ;MAP KERNEL  AND USER PAR'S 4 & 5
042524  010567  127622                  MOV     R5,KIPAR5       ;  BACK TO 12-16K
042530  010567  135114                  MOV     R5,UIPAR4
042534  010567  135112                  MOV     R5,UIPAR5
042540  012767  021276  135502          MOV     #T0250,MMVEC    ;RESTORE ADDR. OF NORMAL M.M. TRAP ROUTINE
042546  000404                          BR      TS376           ;GET TO NEXT TEST
042550  042767  160000  135014  8$:     BIC     #160000,SR0     ;CLEAR ERROR BITS IN SR0
042556  104000                          EMT                     ;M.M. TRAP WHILE IN RELOCATE MODE -
                                                                ;REFERENCED WRONG SET OF PDR'S
                                                                ;FOR TIGHTER SCOPE LOOP
                                                                ;REPLACE ERROR CALL WITH
                                                                ;A "NOP" = 000240

                                ;*************************************************************************
                                ;TEST 376       W-BIT LOGIC TEST, KERNEL PDR'S
                                ;*************************************************************************
042560                          TS376:
042560                          1$:
042560  004767  005300                  JSR     PC,TOFF         ;TURN T-BIT TRAPPING OFF FOR THIS TEST
042564  012702  000004                  MOV     #4,R2           ;SET LOOP COUNTER TO 4
042570  012700  172346                  MOV     #KIPAR3,R0      ;LOAD ADDRESS OF PAR3 INTO R0
042574  012701  001400                  MOV     #1400,R1        ;LOAD "24-28K" PAR VALUE INTO R1
042600  010120                  2$:     MOV     R1,(R0)+        ;MAP PARS 3-6 TO 12-16K
042602  077202                          SOB     R2,2$           ;LOOP TIL ALL 4 OF THEM LOADED
042604  012705  172300                  MOV     #KIPDR0,R5      ;LOAD ADDRESS OF FIRST PDR TO BE TESTED IN R5
042610  012704  000010                  MOV     #10,R4          ;SET LOOP COUNTER TO 8
042614  012703  017776                  MOV     #17776,R3       ;INITIALIZE VIRTUAL ADDRESS TO BE IN R3
```

```
042620  012700  172300          3$:     MOV     #KIPDR0,R0      ;LOAD ADDR. OF FIRST PDR TO BE SETUP IN R0
042624  012702  000010                  MOV     #10,R2          ;SET LOOP COUNTER TO 8
042630  012701  077406                  MOV     #77406,R1       ;PUT 'W-BIT OFF DATA'' INTO R1
042634  010120                  4$:     MOV     R1,(R0)+        ;CLEAR ALL W-BITS BY WRITING TO ALL PDRS
042636  077202                          SOB     R2,4$           ;LOOP UNTIL ALL OF THEM SETUP
042640  011313                          MOV     (R3),(R3)       ;DO 'DATO'' TO VIRTUAL ADDR.-SETTING A W-BIT
042642  031527  000100                  BIT     (R5),#WBIT      ;DID THAT CAUSE W-BIT TO BE SET?
042646  001001                          BNE     5$
042650  104000                          EMT                     ;W-BIT DID NOT GET SET IN PDR
                                                                ;FOR TIGHTER SCOPE LOOP
                                                                ;REPLACE ERROR CALL WITH
                                                                ;'BR 3$'' = 000763
042652  012702  000010          5$:     MOV     #10,R2          ;SET LOOP COUNTER TO 8
042656  012700  172300                  MOV     #KIPDR0,R0      ;LOAD ADDR. OF FIRST PDR TO BE CHECKED IN R0
042662  031027  000100          6$:     BIT     (R0),#WBIT      ;DID W-BIT IN OTHER PDRS REMAIN CLEAR?
042666  001403                          BEQ     7$              ;BRANCH IF YES
042670  020500                          CMP     R5,R0           ;IF W-BIT SET, THEN WAS IT PDR UNDER TEST?
042672  001401                          BEQ     7$
042674  104000                          EMT                     ;W-BIT GOT SET IN MORE THAN ONE PDR
                                                                ;FOR TIGHTER SCOPE LOOP
                                                                ;REPLACE ERROR CALL WITH
                                                                ;'BR 3$'' = 000750
042676  062700  000002          7$:     ADD     #2,R0           ;POINT R0 TO NEXT PDR TO BE CHECKED
042702  077211                          SOB     R2,6$           ;LOOP UNTIL ALL 8 CHECKED FOR CLEAR W-BIT
042704  010115                          MOV     R1,(R5)         ;WRITE TO THE PDR TESTED TO CLEAR W-BIT
042706  031527  000100                  BIT     (R5),#WBIT      ;DID WRITING PDR CLEAR THE W-BIT?
042712  001401                          BEQ     8$
042714  104000                          EMT                     ;W-BIT DID NOT CLEAR BY WRITING THE PDR
                                                                ;FOR TIGHTER SCOPE LOOP
                                                                ;REPLACE ERROR CALL WITH
                                                                ;'BR 3$'' = 000740
042716  062705  000002          8$:     ADD     #2,R5           ;POINT R5 TO THE NEXT PDR TO BE TESTED
042722  062703  020000                  ADD     #20000,R3       ;CHANGE VIRT. ADDR TO REF. NEXT PDR
042726  077444                          SOB     R4,3$           ;LOOP BACK TO 3$ UNTIL ALL 8 PDR'S TESTED
042730  004767  005164                  JSR     PC,TON          ;TURN T-BIT BACK ON FOR NEXT TEST

                                ;*****************************************************************************
                                ;TEST 377       W-BIT LOGIC TEST, USER PDR'S
                                ;*****************************************************************************
042734                          TS377:
042734  012767  140000  135034  1$:     MOV     #140000,PSW     ;GO TO USER MODE FOR THIS TEST
042742  004767  005116                  JSR     PC,TOFF         ;TURN T-BIT TRAPPING OFF FOR THIS TEST
042746  012702  000004                  MOV     #4,R2           ;SET LOOP COUNTER TO 4
042752  012700  177646                  MOV     #UIPAR3,R0      ;LOAD ADDRESS OF PAR3 INTO R0
042756  012701  001400                  MOV     #1400,R1        ;LOAD "24-28K'' PAR VALUE INTO R1
042762  010120                  2$:     MOV     R1,(R0)+        ;MAP PARS 3-6 TO 12-16K
042764  077202                          SOB     R2,2$           ;LOOP TIL ALL 4 OF THEM LOADED
042766  012705  177600                  MOV     #UIPDR0,R5      ;LOAD ADDRESS OF FIRST PDR TO BE TESTED IN R5
042772  012704  000010                  MOV     #10,R4          ;SET LOOP COUNTER TO 8
042776  012703  017776                  MOV     #17776,R3       ;INITIALIZE VIRTUAL ADDRESS TO BE IN R3
043002  012700  177600          3$:     MOV     #UIPDR0,R0      ;LOAD ADDR. OF FIRST PDR TO BE SETUP IN R0
043006  012702  000010                  MOV     #10,R2          ;SET LOOP COUNTER TO 8
043012  012701  077406                  MOV     #77406,R1       ;PUT 'W-BIT OFF DATA'' INTO R1
043016  010120                  4$:     MOV     R1,(R0)+        ;CLEAR ALL W-BITS BY WRITING TO ALL PDRS
043020  077202                          SOB     R2,4$           ;LOOP UNTIL ALL OF THEM SETUP
043022  011313                          MOV     (R3),(R3)       ;DO 'DATO'' TO VIRTUAL ADDR.-SETTING A W-BIT
```

```
043024  031527  000100           BIT     (R5),#WBIT      ;DID THAT CAUSE W-BIT TO BE SET?
043030  001001                   BNE     5$
043032  104000                   EMT                     ;W-BIT DID NOT GET SET IN PDR
                                                         ;FOR TIGHTER SCOPE LOOP
                                                         ;REPLACE ERROR CALL WITH
                                                         ;"BR 3$" = 000763
043034  012702  000010   5$:     MOV     #10,R2          ;SET LOOP COUNTER TO 8
043040  012700  177600           MOV     #UIPDR0,R0      ;LOAD ADDR. OF FIRST PDR TO BE CHECKED IN R0
043044  031027  000100   6$:     BIT     (R0),#WBIT      ;DID W-BIT IN OTHER PDRS REMAIN CLEAR?
043050  001403                   BEQ     7$              ;BRANCH IF YES
043052  020500                   CMP     R5,R0           ;IF W-BIT SET, THEN WAS IT PDR UNDER TEST?
043054  001401                   BEQ     7$
043056  104000                   EMT                     ;W-BIT GOT SET IN MORE THAN ONE PDR
                                                         ;FOR TIGHTER SCOPE LOOP
                                                         ;REPLACE ERROR CALL WITH
                                                         ;"BR 3$" = 000750
043060  062700  000002   7$:     ADD     #2,R0           ;POINT R0 TO NEXT PDR TO BE CHECKED
043064  077211                   SOB     R2,6$           ;LOOP UNTIL ALL 8 CHECKED FOR CLEAR W-BIT
043066  010115                   MOV     R1,(R5)         ;WRITE TO THE PDR TESTED TO CLEAR W-BIT
043070  031527  000100           BIT     (R5),#WBIT      ;DID WRITING PDR CLEAR THE W-BIT?
043074  001401                   BEQ     8$
043076  104000                   EMT                     ;W-BIT DID NOT CLEAR BY WRITING THE PDR
                                                         ;FOR TIGHTER SCOPE LOOP
                                                         ;REPLACE ERROR CALL WITH
                                                         ;"BR 3$" = 000740
043100  062705  000002   8$:     ADD     #2,R5           ;POINT R5 TO THE NEXT PDR TO BE TESTED
043104  062703  020000           ADD     #20000,R3       ;CHANGE VIRT. ADDR TO REF. NEXT PDR
043110  077444                   SOB     R4,3$           ;LOOP BACK TO 3$ UNTIL ALL 8 PDR'S TESTED
043112  004767  005002           JSR     PC,TON          ;TURN T-BIT BACK ON FOR NEXT TEST
043116  005067  134654           CLR     PSW             ;BACK TO KERNEL MODE BEFORE LEAVING

                         ;*******************************************************************************
                         ;TEST 400       TEST "W-BIT" SPECIAL CASES
                         ;*******************************************************************************
043122                   TS400:

043122  004767  004736   1$:     JSR     PC,TOFF         ;TURN OFF T-BIT TRAPPING FOR THIS TEST
043126  012701  077406           MOV     #77406,R1       ;PUT "W-BIT OFF" VALUE FOR PDR IN R1
043132  010167  127160   2$:     MOV     R1,KIPDR7       ;LOAD KERNEL PDR 7 TO CLEAR W-BIT
043136  016700  134430           MOV     SR0,R0          ;READ PRESENT CONTENTS OF STATUS REG. 0
043142  010067  134424           MOV     R0,SR0          ;WRITE PRESENT CONTENTS OF SR0 BACK TO ITSELF
043146  016702  127144           MOV     KIPDR7,R2       ;READ CONTENTS OF KIPDR7 INTO R2
043152  020102                   CMP     R1,R2           ;WAS W-BIT LEFT CLEARED?
043154  001401                   BEQ     3$
043156  104000                   EMT                     ;W-BIT IN KIPDR7 SET WHEN SR0 WAS WRITTEN TO
                                                         ;FOR TIGHTER SCOPE LOOP
                                                         ;REPLACE ERROR CALL WITH
                                                         ;"BR 2$" = 000765
043160  010167  127130   3       MOV     R1,KIPDR6       ;LOAD KERNEL PDR6 WITH 77406 TO CLEAR W-BIT
043164  012767  043176  134612   MOV     #4$,ERRVEC      ;SET UP LOC. 4 TO 4$ FOR ODD ADDR. ABORT
043172  005037  140000           CLR     @#140000        ;CAUSE TIMEOUT ABORT THRU LOC. 4
043176  012706  001000   4$:     MOV     #KERSTK,KSP     ;RESTORE THE STACK POINTER
043202  016702  127106           MOV     KIPDR6,R2       ;READ KIPDR6 INTO R2
043206  052701  000100           BIS     #100,R1         ;R1-77506
043212  020102                   CMP     R1,R2           ;WAS W-BIT SET?
043214  001401                   BEQ     5$
```

```
043216  104000                                  EMT                             ;W-BIT WAS NOT SET DURING A TIMEOUT ABORT
                                                                                ;FOR TIGHTER SCOPE LOOP
                                                                                ;REPLACE ERROR CALL WITH
                                                                                ;"BR 3$" = 000757
043220  010167  127070                  5$:     MOV     R1,KIPDR6               ;RESTORE KIPDR6 TO 77406
043224  012767  001400  127122                  MOV     #1400,KIPAR6            ;RESTORE KIPAR6 TO 1400
043232  012767  021256  134544                  MOV     #T04,ERRVEC             ;RESTORE NORMAL CPU TRAP ROUTINE TO LOC.4
043240  004767  004654                          JSR     PC,TON                  ;TURN T-BIT TRAPPING BACK ON

                                        ;*************************************************************************
                                        ;*
                                        ;*      THE NEXT THREE (3) TESTS CAUSE MEMORY MANAGEMENT ERRORS
                                        ;*      TO CHECK THE ABILITY OF STATUS REGISTER 0 TO RECORD KT
                                        ;*      ERRORS AND THE ABILITY OF STATUS REGISTER 2 TO LOCK UP THE
                                        ;*      VIRTUAL ADDR. OF THE INSTRUCTION THAT CAUSED THE ERROR.
                                        ;*      THE BITS OF SR2 ARE CHECKED AND BITS <15:13>, <6:5>, AND <3:0>
                                        ;*      ARE CHECKED IN SR0.  SO THE SR0 AND SR2 LOGIC AND THE
                                        ;*      KT ERROR LOGIC ARE CHECKED.
                                        ;*
                                        ;*************************************************************************

                                        ;*************************************************************************
                                        ;TEST 401       NON-RESIDENT ABORT TEST (ACF=0&4)
                                        ;*************************************************************************
043244                                  TS401:

043244  012700  001400                  1$:     MOV     #1400,R0                ;LOAD DATA FOR PAR'S INTO R0
043250  010067  127072                          MOV     R0,KIPAR3               ;MAP KERNEL PAR'S 3&4 TO 24-28K
043254  010067  127070                          MOV     R0,KIPAR4
043260  010067  134362                          MOV     R0,UIPAR3               ;MAP USER PAR'S 3&4 TO 24-28K
043264  010067  134360                          MOV     R0,UIPAR4
043270  012767  077406  127010                  MOV     #77406,KIPDR3           ;MAP KERNEL PDR 3  128 BLKS, READ-WRITE
043276  012767  077406  134302                  MOV     #77406,UIPDR3           ;MAP USER PDR 3  128 BLKS, READ-WRITE
043304  012700  060000                          MOV     #60000,R0               ;LOAD VIRTUAL ADDR. TO REFERENCE PDR3 INTO R0
043310  012701  100000                          MOV     #100000,R1              ;LOAD VIRTUAL ADDR. TO REFERENCE PDR4 INTO R1
043314  012703  100011                          MOV     #100011,R3              ;LOAD R3 WITH WHAT SR0 SHOULD READ - N.R., KERNEL, PG.4
043320  012702  077400                          MOV     #77400,R2               ;LOAD ACF=0 (NON-RESIDENT) PDR VALUE IN R2
043324  012767  043360  134716          2$:     MOV     #5$,MMVEC               ;POINT MEM. MGMT. TRAP VECTOR TO 5$ BELOW
043332  010267  126752                          MOV     R2,KIPDR4               ;LOAD ACF TEST VALUE INTO KIPDR4
043336  010267  134246                          MOV     R2,UIPDR4               ;LOAD ACF TEST VALUE INTO UIPDR4
043342  005010                          3$:     CLR     (R0)                    ;CLEAR PHYS. LOC. 140000 USING PDR3
043344  016767  134426  174010                  MOV     PSW,$TMP0               ;SAVE PSW IN CASE OF ERROR
043352  005211                          4$:     INC     (R1)                    ;TRY TO REF. IT USING PDR4 - SHOULD TRAP TO 5$
043354  001452                                  BEQ     TS402
043356  104000                                  EMT                             ;MEM. MGMT. ABORT DID NOT OCCUR
                                                                                ;FOR TIGHTER SCOPE LOOP
                                                                                ;REPLACE ERROR CALL WITH
                                                                                ;"BR 3$" = 000772
043360  062706  000004                  5$:     ADD     #4,SP                   ;RESTORE STACK POINTER
043364  005710                                  TST     (R0)                    ;DID INSTRUCTION GET ABORTED & NOT EXECUTE
043366  001401                                  BEQ     6$
043370  104000                                  EMT                             ;INSTRUCTION WAS NOT ABORTED, LOC. GOT CHANGED
                                                                                ;FOR TIGHTER SCOPE LOOP
                                                                                ;REPLACE ERROR CALL WITH
                                                                                ;"BR 3$" = 000764
043372  016767  134174  173754          6$:     MOV     SR0,WASSR0              ;READ STATUS REGISTER 0
```

```
043400 016767 134172 173750            MOV     SR2,WASSR2        ;READ STATUS REGISTER 2
043406 020367 173742                   CMP     R3,WASSR0         ;DID SR0 REPORT NON-RESIDENT ERROR CORRECTLY?
043412 001401                          BEQ     7$
043414 104000                          EMT                       ;SR0 DID NOT REPORT NON-RES. ERROR CORRECTLY
                                                                 ;FOR TIGHTER SCOPE LOOP
                                                                 ;REPLACE ERROR CALL WITH
                                                                 ;'BR 3$' = 000752
043416 012704 043352            7$:    MOV     #4$,R4            ;LOAD R4 WITH WHAT SR2 SHOULD READ
043422 020467 173730                   CMP     R4,WASSR2         ;DID SR2 LOCKUP RIGHT VIRTUAL ADDR. (=4$)?
043426 001401                          BEQ     8$
043430 104000                          EMT                       ;SR2 DID NOT LOCK VIRTUAL ADDR. OF NON-RES. ERROR
                                                                 ;FOR TIGHTER SCOPE LOOP
                                                                 ;REPLACE ERROR CALL WITH
                                                                 ;'BR 3$' = 000744
043432 042767 160000 134132     8$:    BIC     #160000,SR0       ;CLEAR THE ERROR BITS IN SR0
043440 032767 140000 173714            BIT     #140000,$TMP0     ;HAS ACF=0&4 BEEN TESTED IN USER YET
043446 001006                          BNE     9$                ;BRANCH IF YES
043450 012703 100151                   MOV     #100151,R3        ;LOAD R3 WITH WHAT SP0 SHOULD READ - N.R., USER, PG.4
043454 012767 140000 134314            MOV     #140000,PSW       ;GO TO USER MODE
043462 000720                          BR      2$                ;REPEAT TEST IN USER MODE
043464 022702 077404            9$:    CMP     #77404,R2         ;HAS ACF=4 BEEN TESTED YET?
043470 001407                          BEQ     10$               ;BRANCH IF YES
043472 012702 077404                   MOV     #77404,R2         ;THEN LOAD ACF=4 (NON-RES) PDR VALUE IN R2
043476 012703 10001                    MOV     #100011,R3        ;LOAD R3 WITH WHAT SR0 SHOULD READ-N.R.,KERNEL,PG. 4
043502 005067 134270                   CLR     PSW               ;GO BACK TO KERNEL MODE
043506 000706                          BR      2$                ;GO BACK & TEST ACF=4 IN SAME MODE
043510 005067 134262            10$:   CLR     PSW               ;GO BACK TO KERNEL MODE BEFORE LEAVING
043514 012767 021276 134526            MOV     #T0250,MMVEC      ;RESTORE ADDRESS OF NORMAL MEMORY
                                                                 ;MANAGEMENT ERROR ROUTINE TO MMVEC

                                ;*****************************************************************************
                                ;TEST 402        READ-ONLY ABORT TEST (ACF=2)
                                ;*****************************************************************************
043522                          TS402:
043522                          1$:                              ;KERNEL & USER PAR'S 3 & 4 AND PDR 3
                                                                 ;ARE SETUP FROM LAST TEST
043522 012700 060000                   MOV     #60000,R0         ;LOAD VIRTUAL ADDR. TO REFERENCE PDR3 INTO R0
043526 012701 100000                   MOV     #100000,R1        ;LOAD VIRTUAL ADDR. TO REFERENCE PDR4 INTO R1
043532 012703 020011                   MOV     #20011,R3         ;LOAD R3 WITH WHAT SR0 SHOULD READ - R/O, KERNEL, PG.4
043536 012702 077402                   MOV     #77402,R2         ;LOAD ACF=2 (READ-ONLY) PDR VALUE IN R2
043542 012767 043574 134500     2$:    MOV     #5$,MMVEC         ;POINT MEM. MGMT. TRAP VECTOR TO 5$ BELOW
043550 010267 126534                   MOV     R2,KIPDR4         ;LOAD ACF=2 INTO KIPDR4
043554 010267 134030                   MOV     R2,UIPDR4         ;LOAD ACF=2 INTO UIPDR4
043560 005010                   3$:    CLR     (R0)              ;CLEAR PHYS. LOC. 140000 USING PDR3
043562 016767 134210 173572            MOV     PSW,$TMP0         ;SAVE PSW IN CASE OF ERROR
043570 005211                   4$:    INC     (R1)              ;TRY TO WRITE USING PDR4 - SHOULD TRAP TO 5$
043572 104000                          EMT                       ;MEM. MGMT. ABORT DID NOT OCCUR
                                                                 ;FOR TIGHTER SCOPE LOOP
                                                                 ;REPLACE ERROR CALL WITH
                                                                 ;'BR 3$' = 000772
043574 062706 000004            5$:    ADD     #4,SP             ;RESTORE STACK POINTER
043600 005710                          TST     (R0)              ;DID INSTRUCTION GET ABORTED & NOT EXECUTE
043602 001401                          BEQ     6$
043604 104000                          EMT                       ;INSTRUCTION WAS NOT ABORTED, LOC. GOT CHANGED
                                                                 ;FOR TIGHTER SCOPE LOOP
                                                                 ;REPLACE ERROR CALL WITH
```

```
                                                                ;'BR 3$' = 000764
043606  016767  133760  173540  6$:     MOV     SR0,WASSR0      ;READ STATUS REG. 0
043614  016767  133756  173534          MOV     SR2,WASSR2      ;READ STATUS REG. 2
043622  020367  173526                  CMP     R3,WASSR0       ;DID SR0 REPORT READ-ONLY ERROR CORRECTLY?
043626  001401                          BEQ     7$
043630  104000                          EMT                     ;SR0 DID NOT REPORT R/O ERROR CORRECTLY
                                                                ;FOR TIGHTER SCOPE LOOP
                                                                ;REPLACE ERROR CALL WITH
                                                                ;'BR 3$' = 000752
043632  012704  043570          7$:     MOV     #4$,R4          ;LOAD R4 WITH WHAT SR2 SHOULD READ
043636  020467  173514                  CMP     R4,WASSR2       ;DID SR2 LOCKUP RIGHT VIRTUAL ADDR. (=4$)?
043642  001401                          BEQ     8$
043644  104000                          EMT                     ;SR2 DID NOT LOCKUP VIRTUAL ADDR. OF R/O ERROR
                                                                ;FOR TIGHTER SCOPE LOOP
                                                                ;REPLACE ERROR CALL WITH
                                                                ;'BR 3$' = 000744
043646  042767  160000  133716  8$:     BIC     #160000,SR0     ;CLEAR THE ERROR BITS IN SR0
043654  032767  140000  173500          BIT     #140000,$TMP0   ;HAS ACF=2 BEEN TESTED IN USER MODE?
043662  001006                          BNE     9$              ;BRANCH IF YES
043664  012703  020151                  MOV     #20151,R3       ;LOAD R3 WITH WHAT SR0 SHOULD READ-R/O, USER, PG.4
043670  012767  140000  134100          MOV     #140000,PSW     ;GO TO USER MODE
043676  000721                          BR      2$              ;REPEAT TEST IN USER MODE
043700  005067  134072          9$:     CLR     PSW             ;GO BACK TO KERNEL MODE BEFORE LEAVING
043704  012767  021276  134336          MOV     #T0250,MMVEC    ;RESTORE ADDRESS OF NORMAL MEMORY
                                                                ;MANAGEMENT ERROR ROUTINE TO MMVEC.

                                ;NOTE:  MACRO MSG31A WAS DELETED AS IT DIDN'T APPLY TO F11.

                                ;*************************************************************************
                                ;*
                                ;*      THE NEXT TWO (2) TESTS WILL BE CHECKING THE PAGE LENGTH
                                ;*      COMPARATORS AND SOME MORE OF THE KT ERROR DETECTION
                                ;*      AND STATUS LOGIC.  THE PAGE LENGTH FIELD (PLF) IN KERNEL
                                ;*      PDR 4 IS VARIED AND FOR EVERY PLF, THREE (3) VIRTUAL
                                ;*      ADDRESSES ARE READ.  WHILE USING BOTH UPWARD & DOWNWARD PAGE
                                ;*      EXPANSION, ONE OF THOSE THREE VIRTUAL ADDRESSES WILL CAUSE A
                                ;*      "PAGE LENGTH ABORT" WHILE THE OTHER TWO WON'T.
                                ;*
                                ;*      STATUS REGISTER 0 & 2 ARE CHECKED WHEN THE PAGE LENGTH
                                ;*      ABORT DOES OCCUR TO SEE THAT THE ABORT IS REPORTED AND THAT
                                ;*      THE VIRTUAL ADDRESS OF THE INSTRUCTION THAT CAUSED THE ABORT
                                ;*      IS LOCKED UP.
                                ;*
                                ;*************************************************************************
```

```
                                        ;*****************************************************************************
                                        ;TEST 403       PAGE LENGTH FAULTS-UPWARD EXPANSION
                                        ;*****************************************************************************
 043712                                 TS403:
 043712  012767  077406  126366         1$:     MOV     #77406,KIPDR3   ;MAKE SURE PDR3 IS DESCRIBED AS R/W
 043720  012767  077406  126364                 MOV     #77406,KIPDR5   ;MAKE SURE PDR5 IS DESCRIBED AS R/W
 043726  012700  044126                         MOV     #DALTB1,R0      ;DAL TABLE FOR VIRTUAL ADDR'S. TO SELECT PDR4.
 043732  012704  044144                         MOV     #PDRTB1,R4      ;PDR TABLE FOR PDR4 (COINCIDES WITH DAL TABLE).
 043736  012701  000006                         MOV     #6,R1           ;SET UP LOOP COUNTER.
 043742  012767  044104  134300                 MOV     #9$,MMVEC       ;SETUP M.M. TRAP VECTOR FOR UNEXPECTED ABORTS
 043750  012706  001000                         MOV     #KERSTK,KSP     ;MAKE SURE STACK POINTER IS ALL SET UP

                                        ;TEST NON-ABORT CASES (VBA < OR = PLF)
 043754  012467  126330                 2$:     MOV     (R4)+,KIPDR4    ;LOAD KIPDR4 WITH PAGE LENGTH VALUE
 043760  005730                                 TST     @(R0)+          ;ACCESS VIRTUAL ADDR. (VBA < OR = PLF)
                                                                        ;NO ABORT SHOULD OCCUR!!!
 043762  077104                                 SOB     R1,2$           ;DONE?...NO- TEST NEXT COMBINATION OF DAL & PDR.

                                        ;TEST ABORT CASES (VBA > PLF)
 043764  012701  000005                 3$:     MOV     #5,R1           ;SET UP LOOP COUNTER.
 043770  012700  044162                         MOV     #DALTB2,R0      ;DAL TABLE
 043774  012704  044176                         MOV     #PDRTB2,R4      ;PDR TABLE
 044000  012767  044020  134242                 MOV     #6$,MMVEC       ;SETUP M.M. TRAP VECTOR FOR EXPECTED ABORT

 044006  012467  126276                 4$:     MOV     (R4)+,KIPDR4    ;LOAD KIPDR4 WITH PAGE LENGHT VALUE
 044012  005730                         5$:     TST     @(R0)+          ;ACCESS VIRTUAL ADDR. (VBA > PLF - ABORT TO 6$)
 044014  001476                                 BEQ     TS404
 044016  104000                                 EMT                     ;EXPECTED PAGE LENGTH ABORT DID NOT OCCUR
                                                                        ;FOR TIGHTER SCOPE LOOP
                                                                        ;REPLACE ERROR CALL WITH
                                                                        ;"BR 5$" = 000776
 044020  012706  001000                 6$:     MOV     #KERSTK,KSP     ;RESTORE STACK POINTER FOLLOWING ABORT
 044024  016767  133542  173322                 MOV     SR0,WASSR0      ;READ M.M. STATUS REG. 0
 044032  016767  133540  173316                 MOV     SR2,WASSR2      ;READ M.M. STATUS REG. 2
 044040  012702  040011                         MOV     #40011,R2       ;PUT EXPECTED SR0 CONTENTS IN R2
 044044  020267  173304                         CMP     R2,WASSR0       ;DID SR0 REPORT PG. LENGTH ABORT, PAGE 4, KERNEL?
 044050  001401                                 BEQ     7$
 044052  104000                                 EMT                     ;SR0 DID NOT REPORT PG. LENGTH ABORT CORRECTLY
                                                                        ;FOR TIGHTER SCOPE LOOP
                                                                        ;REPLACE ERROR CALL WITH
                                                                        ;"BR 5$" = 000757
 044054  012703  044012                 7$:     MOV     #5$,R3          ;PUT EXPECTED SR2 CONTENTS IN R3
 044060  020367  173272                         CMP     R3,WASSR2       ;DID SR2 LOCKUP VIRT. ADDR. OF ABORTED INSTRUCTION?
 044064  001401                                 BEQ     8$
 044066  104000                                 EMT                     ;SR2 DID NOT LOCKUP VIRT. ADDR. OF ABORT CORRECTLY
                                                                        ;FOR TIGHTER SCOPE LOOP
                                                                        ;REPLACE ERROR CALL WITH
                                                                        ;"BR 5$" = 000751
 044070  042767  160000  133474         8$:     BIC     #160000,SR0     ;CLEAR ERROR BITS IN SR0
 044076  077135                                 SOB     R1,4$           ;DONE?..NO - GET NEXT DAL & PDR PAIR
 044100  000167  000010                         JMP     10$             ;YES...
 044104  042767  160000  133460         9$:     BIC     #160000,SR0     ;CLEAR ERROR BITS IN SR0
 044112  104000                                 EMT                     ;GOT PG. LENGTH ABORT BEFORE IT WAS EXPECTED
                                                                        ;FOR TIGHTER SCOPE LOOP
                                                                        ;REPLACE ERROR CALL WITH
                                                                        ;A "NOP" = 240
```

```
044114  012767  021276  134126  10$:    MOV     #T0250,MMVEC    ;RESTORE NORMAL M.M. TRAP HANDLER
                                                                ;ADDRESS TO M.M. TRAP VECTOR
044122  000167  000064                  JMP     TS404           ;GET TO NEXT TEST

                                ;DAL TABLE FOR UPWARD EXPANSION (NON-ABORT CASES)
044126  100000                  DALTB1: 100000
044130  106100                          106100
044132  102300                          102300
044134  102500                          102500
044136  113700                          113700
044140  104600                          104600
044142  117700                          117700

                                ;PDR TABLE FOR KPDR4 (NON-ABORT CASES)
044144  000006                  PDRTB1: 000006
044146  052006                          052006
044150  045006                          045006
044152  052006                          052006
044154  074406                          074406
044156  025006                          025006
044160  077406                          077406

                                ;DAL TABLE (ABORT CASES)
044162  100100                  DALTB2: 100100
044164  110100                          110100
044166  116600                          116600
044170  112700                          112700
044172  117000                          117000
044174  117700                          117700

                                ;PDR TABLE (ABORT CASES)
044176  000006                  PDRTB2: 000006
044200  030406                          030406
044202  046406                          046406
044204  042006                          042006
044206  073406                          073406
044210  077006                          077006


                                ;*******************************************************************************
                                ;TEST 404       PAGE LENGTH FAULTS-DOWNWARD EXPANSION
                                ;*******************************************************************************
044212                          TS404:
044212  012700  044412          1$:     MOV     #DALTB3,R0      ;DAL TABLE FOR VIRTUAL ADDR S. TO SELECT PDR4.
044216  012704  044430                  MOV     #PDRTB3,R4      ;PDR TABLE FOR PDR4 (COINCIDES WITH DAL TABLE).
044222  012701  000006                  MOV     #6,R1           ;SET UP LOOP COUNTER.
044226  012767  044370  134014          MOV     #9$,MMVEC       ;SETUP M.M. TRAP VECTOR FOR UNEXPECTED ABORTS
044234  012706  001000                  MOV     #KERSTK,KSP     ;MAKE SURE STACK POINTER IS ALL SET UP

                                ;TEST NON-ABORT CASES (VBA > OR = PLF)
044240  012467  126044          2$:     MOV     (R4)+,KIPDR4    ;LOAD KIPDR4 WITH PAGE LENGTH VALUE
044244  005730                          TST     @(R0)+          ;ACCESS VIRTUAL ADDR. (VBA > OR = PLF)
                                                                ;NO ABORT SHOULD OCCUR!!!
044246  077104                          SOB     R1,2$           ;DONE?..NO- TEST NEXT COMBINATION OF DAL & PDR.
```

```
                                        ;TEST ABORT CASES (VBA < PLF)
044250  012701  000005          3$:     MOV     #5,R1           ;SET UP LOOP COUNTER.
044254  012700  044446                  MOV     #DALTB4,R0      ;DAL TABLE
044260  012704  044462                  MOV     #PDRTB4,R4      ;PDR TABLE
044264  012767  044304  133756          MOV     #6$,MMVEC       ;SETUP M.M. TRAP VECTOR FOR EXPECTED ABORT

044272  012467  126012          4$:     MOV     (R4)+,KIPDR4    ;LOAD KIPDR4 WITH PAGE LENGHT VALUE
044276  005730                  5$:     TST     @(R0)+          ;ACCESS VIRTUAL ADDR. (VBA < PLF - ABORT TO 6$)
044300  001476                          BEQ     TS405
044302  104000                          EMT                     ;EXPECTED PAGE LENGTH ABORT DID NOT OCCUR
                                                                ;FOR TIGHTER SCOPE LOOP
                                                                ;REPLACE ERROR CALL WITH
                                                                ;"BR 5$" = 000776
044304  012706  001000          6$:     MOV     #KERSTK,KSP     ;RESTORE STACK POINTER FOLLOWING ABORT
044310  016767  133256  173036          MOV     SR0,WASSR0      ;READ M.M. STATUS REG. 0
044316  016767  133254  173032          MOV     SR2,WASSR2      ;READ M.M. STATUS REG. 2
044324  012702  040011                  MOV     #40011,R2       ;PUT EXPECTED SR0 CONTENTS IN R2
044330  020267  173020                  CMP     R2,WASSR0       ;DID SR0 REPORT PG. LENGTH ABORT, PAGE 4, KERNEL?
044334  001401                          BEQ     7$
044336  104000                          EMT                     ;SR0 DID NOT REPORT PG. LENGTH ABORT CORRECTLY
                                                                ;FOR TIGHTER SCOPE LOOP
                                                                ;REPLACE ERROR CALL WITH
                                                                ;"BR 5$" = 000757
044340  012703  044276          7$:     MOV     #5$,R3          ;PUT EXPECTED SR2 CONTENTS IN R3
044344  020367  173006                  CMP     R3,WASSR2       ;DID SR2 LOCKUP VIRT. ADDR. OF ABORTED INSTRUCTION?
044350  001401                          BEQ     8$
044352  104000                          EMT                     ;SR2 DID NOT LOCKUP VIRT. ADDR. OF ABORT CORRECTLY
                                                                ;FOR TIGHTER SCOPE LOOP
                                                                ;REPLACE ERROR CALL WITH
                                                                ;"BR 5$" = 000751
044354  042767  160000  133210  8$:     BIC     #160000,SR0     ;CLEAR ERROR BITS IN SR0
044362  077135                          SOB     R1,4$           ;DONE?..NO - GET NEXT DAL & PDR PAIR
044364  000167  000010                  JMP     10$             ;YES...
044370  042767  160000  133174  9$:     BIC     #160000,SR0     ;CLEAR ERROR BITS IN SR0
044376  104000                          EMT                     ;GOT PG. LENGTH ABORT BEFORE IT WAS EXPECTED
                                                                ;FOR TIGHTER SCOPE LOOP
                                                                ;REPLACE ERROR CALL WITH
                                                                ;A "NOP" = 000240

044400  012767  021276  133642  10$:    MOV     #T0250,MMVEC    ;RESTORE NORMAL M.M. TRAP HANDLER
                                                                ;ADDRESS TO M.M. TRAP VECTOR
044406  000167  000064                  JMP     TS405           ;GET TO NEXT TEST

                                        ;DAL TABLE FOR DOWNWARD EXPANSION (NON-ABORT CASES)
044412  117700                  DALTB3: 117700
044414  111600                          111600
044416  115400                          115400
044420  115200                          115200
044422  104000                          104000
044424  113100                          113100
044426  100000                          100000

                                        ;PDR TABLE (NON-ABORT CASES)
044430  077416                  PDRTB3: 77416
044432  025416                          25416
044434  032416                          32416
```

```
044436  025416                                      25416
044440  003016                                      03016
044442  052416                                      52416
044444  000016                                      00016

                                    ;DAL TABLE (ABORT CASES)
044446  117600                      DALTB4: 117600
044450  107600                              107600
044452  101100                              101100
044454  105000                              105000
044456  100700                              100700
044460  100000                              100000

                                    ;PDR TABLE (ABORT CASES)
044462  077416                      PDRTB4: 77416
044464  047016                              47016
044466  031016                              31016
044470  035416                              35416
044472  004016                              04016
044474  000416                              00416



                                    ;*************************************************************************
                                    ;TEST 405          SR2 BIT TEST
                                    ;*************************************************************************
044476                              TS405:
044476  012767  001400  125642      1$:     MOV     #1400,KIPAR3     ;BE SURE PAR3 IS MAPPED TO 24-28K
044504  012767  001400  125636              MOV     #1400,KIPAR4     ;BE SURE PAR4 IS MAPPED TO 24-28K
044512  012767  077406  125566              MOV     #77406,KIPDR3    ;MAP PAGE 3  128 BLOCKS, R/W
044520  012767  077402  125562              MOV     #77402,KIPDR4    ;MAP PAGE 4  128 BLOCKS, READ-ONLY
044526  012700  060002                      MOV     #60002,R0        ;LOAD R0 WITH VIRTUAL ADDR. WHICH USES PDR3
044532  012701  100002                      MOV     #100002,R1       ;LOAD R1 WITH VIRTUAL ADDR. WHICH USES PDR4
044536  012767  044564  133504              MOV     #3$,MMVEC        ;SET M.M. TRAP VECTOR TO 3$
044544  012720  010727              2$:     MOV     #010727,(R0)+    ;LOAD "MOV PC,(PC)+" INSTRUCTION AT ADDR.
044550  005020                              CLR     (R0)+            ;        REACHED THRU PDR/PAR 4.
044552  012720  000137                      MOV     #000137,(R0)+    ;LOAD "JMP @#3$" INSTRUCTION AT VIRT. ADDR.
044556  012710  044564                      MOV     #3$,(R0)         ;        IN CASE R/O VIOL. DOES NOT ABORT
044562  010107                              MOV     R1,PC            ;TRANSFER PROGRAM EXECUTION TO "PAGE 4 INSTRUCTIONS"
044564  012706  001000              3$:     MOV     #KERSTK,KSP      ;RESTORE STACK POINTER
044570  016767  133002  172560              MOV     SR2,WASSR2       ;READ CONTENTS OF STATUS REG 2
044576  020167  172554                      CMP     R1,WASSR2        ;WAS ADDR. OF "RELOCATED - R/O ABORT" LOCKED UP?
044602  001401                              BEQ     4$
044604  104000                              EMT                      ;SR2 DID NOT LOCK UP VIRTUAL ADDR. OF R/O VIOL.
                                                                     ;FOR TIGHTER SCOPE LOOP
                                                                     ;REPLACE ERROR CALL WITH
                                                                     ;"BR 2$" = 000757
044606  042767  160000  132756      4$:     BIC     #160000,SR0      ;CLEAR THE ERROR BITS IN SR0
044614  060101                              ADD     R1,R1            ;SETUP TO FORM NEXT VIRTUAL ADDRESS
044616  010100                              MOV     R1,R0            ;SETUP R0 TO FORM NEXT VIRT. ADDR. TO LOAD
044620  052701  100000                      BIS     #100000,R1       ;FORM VIRTUAL ADDR. THAT SHOULD BE LOCKED UP NEXT
044624  052700  060000                      BIS     #60000,R0        ;POINT R0 TO NEXT VIRT. ADDR. TO LOAD
044630  020127  110000                      CMP     R1,#110000       ;HAVE ALL VBA'S 100000-110000 BEEN TESTED?
044634  101743                              BLOS    2$               ;BRANCH IF NO

044636  012767  077406  125444              MOV     #77406,KIPDR4    ;RESTORE PDR4 TO R/W ACCESS
```

```
044644  012767  021276  133376          MOV     #T0250,MMVEC        ;RESTORE ADDRESS OF NORMAL M.M.
                                                                    ;TRAP HANDLER TO M.M. VECTOR


                                ;*******************************************************************************
                                ;TEST 406        MORE CHECKS OF SR0 & SR2
                                ;*******************************************************************************
044652                          TS406:
044652                          1$:
044652  032737  000001  001020          BIT     #1, @#$ENV          ;ARE WE RUNNING UNDER APT
044660  001403                          BEQ     70$                 ;IF NO THEN DO TEST
044662  005737  001006                  TST     @#$PASS             ;IS THIS FIRST PASS
044666  001166                          BNE     TS407               ;IF NO THEN SHIP TO NEXT TEST
044670                          70$:
044670  012767  001400  125454          MOV     #1400,KIPAR5        ;MAP KERNEL PAGE 5 TO 24-28K
044676  012767  000406  125404          MOV     #406,KIPDR4         ;SETUP PDR4 FOR PAGE LENGTH ABORT
044704  012767  077402  125400          MOV     #77402,KIPDR5       ;SETUP PDR5 FOR R/O ABORT
044712  016767  132660  172436  2$:     MOV     SR2,WASSR2          ;READ SR2 TO SEE IF ITS TRACKING
044720  012701  044712                  MOV     #2$,R1              ;PUT EXPECTED VIRTUAL PC IN R1
044724  020167  172426                  CMP     R1,WASSR2           ;DID SR2 CONTAIN VIRTUAL PC AT 2$?
044730  001401                          BEQ     4$
044732  104000                          EMT                         ;SR2 NOT TRACKING CORRECTLY
                                                                    ;FOR TIGHTER SCOPE LOOP
                                                                    ;REPLACE ERROR CALL WITH
                                                                    ;'BR 2$" = 000767
044734  016767  132636  172414  4$:     MOV     SR2,WASSR2          ;READ SR2 TO SEE IF ITS TRACKING
044742  012701  044734                  MOV     #4$,R1              ;PUT EXPECTED VIRTUAL PC IN R1
044746  020167  172404                  CMP     R1,WASSR2           ;DID SR2 CONTAIN VIRTUAL PC AT 4$
044752  001401                          BEQ     6$
044754  104000                          EMT                         ;SR2 NOT TRACKING CORRECTLY
                                                                    ;FOR TIGHTER SCOPE LOOP
                                                                    ;REPLACE ERROR CALL WITH
                                                                    ;'BR 4$" = 000767
044756  012767  044774  133264  6$:     MOV     #7$,MMVEC           ;PUT ADDRESS OF 7$ IN M.M. TRAP VECTOR
044764  005067  172374                  CLR     $TMP1               ;CLEAR ERROR INDICATOR
044770  005237  100500                  INC     @#100500            ;CAUSE PAGE LENGTH ABORT - TRAP TO 7$
044774  012706  001000          7$:     MOV     #KERSTK,KSP         ;RESTORE STACK POINTER AFTER ABORT
045000  016767  132566  172354          MOV     SR0,$TMP0           ;SAVE SR0'S INFORMATION ON PG. LGTH. ABORT
045006  016767  132564  172352          MOV     SR2,$TMP2           ;SAVE SR2'S INFORMATION ON PG. LGTH. ABORT
045014  012767  045026  133226          MOV     #8$,MMVEC           ;PUT ADDRESS OF 8$ IN M.M. TRAP VECTOR
045022  005237  120000                  INC     @#120000            ;CAUSE R/O ABORT - TRAP TO 8$
045026  012706  001000          8$:     MOV     #KERSTK,KSP         ;RESTORE STACK POINTER AFTER ABORT
045032  016767  132534  172314          MOV     SR0,WASSR0          ;READ SR0 FOLLWOING SECOND KT ABORT
045040  016767  132532  172310          MOV     SR2,WASSR2          ;READ SR2 FOLLOWING SECOND KT ABORT
045046  026767  172310  172300          CMP     $TMP0,WASSR0        ;IS SR0 STILL HOLDING INFO ON FIRST ABORT?
045054  001402                          BEQ     9$                  ;BRANCH IF YES
045056  005267  172302                  INC     $TMP1               ;SET ERROR INDICATOR
045062  026767  172300  172266  9$:     CMP     $TMP2,WASSR2        ;DOES SR2 STILL HOLD PC OF FIRST ABORT?
045070  001402                          BEQ     10$                 ;BRANCH IF YES
045072  005267  172266                  INC     $TMP1               ;SET ERROR INDICATOR
045076  005767  172262          10$:    TST     $TMP1               ;WERE SR0 OR SR2 CHANGED BY A SECOND ABORT?
045102  001401                          BEQ     11$
045104  104000                          EMT                         ;ONE OF STATUS REGS. CHANGED BY SECOND ABORT
                                                                    ;FOR TIGHTER SCOPE LOOP
                                                                    ;REPLACE ERROR CALL WITH
                                                                    ;'BR 6$" = 000726
```

```
045106  005067  172252          11$:    CLR     $TMP1           ;CLEAR ERROR INDICATOR
045112  000005                          RESET                   ;EXECUTE A RESET, APPLYING AN "INIT"
045114  005067  132452                  CLR     SR0
045120  016767  132446  172226          MOV     SR0,WASSR0      ;READ SR0
045126  005767  172222                  TST     WASSR0          ;WAS SR0 CLEARED BY THE RESET?
045132  001402                          BEQ     12$             ;BRANCH IF YES
045134  005267  172224                  INC     $TMP1           ;SR0 NOT CLEARED BY A RESET
045140  016767  132432  172210  12$:    MOV     SR2,WASSR2      ;READ SR2
045146  022767  045140  172202          CMP     #12$,WASSR2     ;WAS SR2 UNLOCKED BY A RESET?
045154  001402                          BEQ     13$             ;BRANCH IF YES
045156  005267  172202                  INC     $TMP1           ;SR2 NOT UNLOCKED BY A RESET
045162  005767  172176          13$:    TST     $TMP1           ;WERE SR0 & SR2 BOTH "RESET" BY A RESET?
045166  001401                          BEQ     14$
045170  104000                          EMT                     ;SR0 OR SR2 NOT "RESET" BY A RESET
                                                                ;FOR TIGHTER SCOPE LOOP
                                                                ;REPLACE ERROR CALL WITH
                                                                ;"BR 6$" = 000676
045172  012767  000001  132372  14$:    MOV     #1,SR0          ;TURN MEMORY MANAGEMENT BACK ON
045200  016767  132372  172150  15$:    MOV     SR2,WASSR2      ;READ SR2 TO SEE IF ITS TRACKING AGAIN
045206  012701  045200                  MOV     #15$,R1         ;PUT EXPECTED VIRTUAL PC IN R1
045212  020167  172140                  CMP     R1,WASSR2       ;DID SR2 CONTAIN VIRTUAL PC AT 15$
045216  001401                          BEQ     16$
045220  104000                          EMT                     ;SR2 NOT TRACKING CORRECTLY
                                                                ;FOR TIGHTER SCOPE LOOP
                                                                ;REPLACE ERROR CALL WITH
                                                                ;"BR 6$" = 000663
045222  012767  077406  125060  16$:    MOV     #77406,KIPDR4   ;RESET PDR4 TO 128 BLKS, R/W
045230  012767  077406  125054          MOV     #77406,KIPDR5   ;RESET PDR5 TO 128 BLKS, R/W
045236  012767  021276  133004          MOV     #T0250,MMVEC    ;RESTORE ADDRESS OF NORMAL MEMORY
                                                                ;MANAGEMENT TRAP ROUTINE TO M.M. VECTOR



                                ;*************************************************************************
                                ;TEST 407       USER ABORT PICKS UP KERNEL SPACE VECTOR
                                ;*************************************************************************
045244                          TS407:
045244  004767  002614          1$:     JSR     PC,TOFF         ;TURN OFF T-BIT TRAPPING FOR THIS TEST
045250  005067  132522          2$:     CLR     PSW             ;GO TO KERNEL MODE
045254  012706  001000                  MOV     #KERSTK,KSP     ;SETUP KERNEL STACK PTR.
045260  012767  001400  132352          MOV     #1400,UIPAR0    ;MAP USER PAGE 0 TO 24K
045266  012737  045336  000004          MOV     #4$,@#4         ;LOAD KERNEL VECTOR 4 (LOC.4) WITH 4$
045274  012737  000340  000006          MOV     #340,@#6        ;LOAD VECTOR+2 WITH NEW PSW
045302  012767  140000  132466          MOV     #140000,PSW     ;GO TO USER MODE
045310  012706  000600                  MOV     #USESTK,USP     ;SETUP USER STACK PTR.
045314  012737  045334  000004          MOV     #3$,@#4         ;LOAD USER VECTOR 4 (LOC. 60004) WITH 3$
045322  012737  000340  000006          MOV     #340,@#6        ;LOAD VECTOR+2 WITH NEW PSW
045330  005767  112444                  TST     160000          ;CAUSE TIMEOUT ERROR TRAP TO "4"
                                                                ;SHOULD PICK UP NEW PC-4$ FROM KERNEL
                                                                ;LOC. 4, NOT PC=3$ FROM USER LOC. 4 (=60004)
045334                          3$:
045334  104000                          EMT                     ;DID NOT TRAP THRU KERNEL SPACE
                                                                ;FOR TIGHTER SCOPE LOOP
                                                                ;REPLACE ERROR CALL WITH
                                                                ;"BR 2$" = 000740
045336  005067  132434          4$:     CLR     PSW             ;BE SURE BACK IN KERNEL MODE
045342  012706  001000                  MOV     #KERSTK,KSP     ;RESTORE KERNEL S.P. IN CASE IT CHANGED
```

```
045346  005067  132266                      CLR     UIPAR0          ;REMAP USER PAGE 0 TO 0-4K
045352  012767  140000  132416              MOV     #140000,PSW     ;GO TO USER MODE
045360  012706  000600                      MOV     #USESTK,USP     ;RESTORE USER STACK POINTER
045364  005067  132406                      CLR     PSW             ;GO BACK TO KERNEL MODE
045370  012737  021256  000004              MOV     #T04,@#4        ;RESTORE ADDR. OF NORMAL CPU TRAP HANDLER TO 4
045376  004767  002516                      JSR     PC,TON          ;TURN T-BIT TRAPPING BACK ON

                                    ;*************************************************************************
                                    ;TEST 410       RTI IN USER MODE DOES NOT CHANGE PSW
                                    ;*************************************************************************
045402                              TS410:

045402  012702  170000                      MOV     #170000,R2      ;LOAD "PRESENT & EXPECTED" PSW VALUE INTO R2
045406  010267  132364              2$:     MOV     R2,PSW          ;GO TO USER MODE-PRIORITY 0
045412  012746  000340                      MOV     #340,-(SP)      ;PUT A NEW PSW (PRIORITY=7) ON STACK
045416  012746  045424                      MOV     #3$,-(SP)       ;PUT NEW PC ON THE STACK
045422  000002                              RTI                     ;DO AN RTI FROM USER MODE
045424  016701  132346              3$:     MOV     PSW,R1          ;READ NEW PSW INTO R1
045430  042701  007437                      BIC     #7437,R1        ;MASK OFF COND. CODE, T-BIT, AND UNUSED BITS
045434  005067  132336                      CLR     PSW             ;GO BACK TO KERNEL MODE
045440  020201                              CMP     R2,R1           ;DID PSW STAY IN USER, PRIORITY=0
045442  001401                              BEQ     TS411
045444  104000                              EMT                     ;PSW CHANGED BY AN RTI FROM USER
                                                                    ;FOR A TIGHTER SCOPE LOOP
                                                                    ;REPLACE ERROR CALL WITH
                                                                    ;"BR=2$" = 000760



                                    ;*************************************************************************
                                    ;TEST 411       KT ERROR SERVICED BEFORE TIMEOUT ERROR
                                    ;*************************************************************************
045446                              TS411:
045446  012705  077006              1$:     MOV     #77006,R5       ;LOAD PDR7 DATA INTO R5
045452  010567  124640                      MOV     R5,KIPDR7       ;MAP PAGE 7 R/W PLF=176
045456  012737  045476  000004              MOV     #3$,@#4         ;SET CPU TRAP VECTOR TO ADDRESS OF 3$
045464  012737  045500  000250              MOV     #4$,@#250       ;SET M.M. TRAP VECTOR TO ADDRESS OF 4$
045472  005237  177700              2$:     INC     @#177700        ;CAUSE PLF ABORT AND POTENTIAL TIMEOUT
045476                              3$:
045476  104000                              EMT                     ;TRAPPED THRU CPU TRAP VECTOR BUT SHOULDN'T HAVE
                                                                    ;FOR TIGHTER SCOPE LOOP
                                                                    ;REPLACE ERROR CALL WITH
                                                                    ;"BR 2$" = 000776
045500  012706  001000              4$:     MOV     #KERSTK,KSP     ;RESTORE STACK POINTER AFTER TRAPPING
045504  016767  132062  171642              MOV     SR0,WASSR0      ;READ STATUS REG.0
045512  016767  132060  171636      5$:     MOV     SR2,WASSR2      ;READ STATUS REG. 2
045520  012700  040017                      MOV     #40017,R0       ;LOAD EXPECTED SR0 CONTENTS INTO R0
045524  020067  171624                      CMP     R0,WASSR0       ;SR0 PLF ERROR BIT SET?
045530  001401                              BEQ     6$
045532  104000                              EMT                     ;SR0 DIDN'T REPORT PLF ERROR
                                                                    ;FOR TIGHTER SCOPE LOOP
                                                                    ;REPLACE ERROR CALL WITH
                                                                    ;"BR 2$" = 000741
045534  012701  045472              6$:     MOV     #2$,R1          ;LOAD EXPECTED SR2 CONTENTS INTO R1
045540  020167  171612                      CMP     R1,WASSR2       ;WAS SR2 LOCKED BY PLF ABORT?
045544  001401                              BEQ     7$
045546  104000                              EMT                     ;SR2 DIDN'T LOCK UP VIRTUAL ADDRESS
```

```
                                                              ;FOR TIGHTER SCOPE LOOP
                                                              ;REPLACE ERROR CALL WITH
                                                              ;"BR 2$" = 000741
045550  042767  160000  132014  7$:   BIC   #160000,SR0       ;CLEAR ERROR BITS THAT WERE SET IN SR0
045556  012737  021256  000004        MOV   #T04,@#4          ;RESTORE ADDRESS OF NORMAL CPU TRAP HANDLER
045564  012737  021276  000250        MOV   #T0250,@#250      ;RESTORE ADDRESS OF NORMAL M.M. TRAP HANDLER
045572  012767  077406  124516        MOV   #77406,KIPDR7     ;REMAP PAGE 7 TO READ/WRITE PLF=177


                                      ;*******************************************************************************
                                      ;TEST 412       PC & PSW SAVED FOR KT ERROR DURING SERVICE OF TIMEOUT ERROR
                                      ;*******************************************************************************
045600                                TS412:
045600  004767  002260          1$:   JSR   PC,TOFF           ;TURN T-BIT TRAPPING OFF FOR THIS TEST
045604  012767  001400  132034        MOV   #1400,UIPAR3      ;MAP USER PAGE 3 TO 24-28K
045612  012767  001400  132030        MOV   #1400,UIPAR4      ;MAP USER PAGE 4 TO 24-28K
045620  012767  077402  131760        MOV   #77402,UIPDR3     ;MAP USER PAGE 3 READ-ONLY
045626  012767  077406  131754        MOV   #77406,UIPDR4     ;MAP USER PAGE 4 READ/WRITE
045634  012737  045702  000004        MOV   #4$,@#4           ;LOAD ADDRESS OF 4$ IN CPU (TIMEOUT) VECTOR
045642  012737  140017  000006        MOV   #140017,@#6       ;LOAD PSW THAT SHOULD BE PUT ON STACK IN VECTOR+2
045650  012737  045702  000250        MOV   #4$,@#250         ;LOAD ADDRESS OF 4$ IN M.M. TRAP VECTOR
045656  012737  000340  000252        MOV   #340,@#252        ;LOAD A KERNEL PSW IN MMVEC+2
045664  012767  140000  132104  2$:   MOV   #140000,PSW       ;GO TO USER MODE
045672  012706  100002                MOV   #100002,USP       ;SET USER STACK PTR. SO SECOND PUSH IS IN PG. 3
045676  005737  177700          3$:   TST   @#177700          ;CAUSE TIMEOUT ERROR THAT WILL CAUSE
                                                              ;R/O ERROR WHEN TRY TO SAVE OLD PC
045702  016601  000002          4$:   MOV   2(KSP),R1         ;PUT PSW SAVED ON KERNEL STACK INTO R1
045706  011603                        MOV   (KSP),R3          ;PUT PC SAVED ON KERNEL STACK INTO R3
045710  016767  131656  171436        MOV   SR0,WASSR0        ;READ THE CONTENTS OF M.M. STATUS REG. 0
045716  016767  131654  171432        MOV   SR2,WASSR2        ;READ THE CONTENTS OF M.M. STATUS REG. 2
045724  042767  160000  131640        BIC   #160000,SR0       ;CLEAR THE ERROR BITS IN SR0
045732  005067  132040                CLR   PSW               ;BE SURE IN KERNEL MODE
045736  012706  001000                MOV   #KERSTK,KSP       ;RESTORE KERNEL STACK POINTER
045742  012767  140000  132026        MOV   #140000,PSW       ;GO TO USER MODE
045750  012706  000600                MOV   #USESTK,USP       ;RESTORE USER STACK POINTER
045754  005067  132016                CLR   PSW               ;GO BACK TO KERNEL MODE
045760  005067  171376                CLR   $TMP0             ;CLEAR ERROR INDICATOR
045764  020127  170017                CMP   R1,#170017        ;WAS THE PSW SAVED THE ONE PICKED UP BY THE
                                                              ;TIMEOUT TRAP FROM ERRVEC+2?
                                                              ;VALUE 170017 = PSW FROM LOC. 6 WITH
                                                              ;PREVIOUS MODE BITS = USER
045770  001402                        BEQ   5$                ;BRANCH IF YES
045772  005267  171364                INC   $TMP0             ;WRONG PSW SAVED DURING "DOUBLE ERROR" SEQUENCE
045776  020327  045702          5$:   CMP   R3,#3$+4          ;WAS THE PC AT THE TIME OF THE TIMEOUT ERROR
                                                              ;SAVED ON THE STACK?
046002  001402                        BEQ   6$                ;BRANCH IF YES
046004  005267  171352                INC   $TMP0             ;WRONG PC SAVED DURING TRAP SEQUENCE
046010  026727  171340  020147  6$:   CMP   WASSR0,#20147     ;DID SR0 REPORT - USER, PAGE 3, R/O ABORT?
046016  001402                        BEQ   7$                ;BRANCH IF YES
046020  005267  171336                INC   $TMP0             ;SR0 DID NOT REPORT R/O ABORT
046024  026727  171326  045676  7$:   CMP   WASSR2,#3$        ;DID SR2 LOCK UP VIRTUAL ADDR. OF LAST
                                                              ;INSTRUCTION SUCCESSFULLY FETCHED?
046032  001402                        BEQ   8$                ;BRANCH IF YES
046034  005267  171322                INC   $TMP0             ;SR2 DID NOT LOCK UP ADDR. OF TIMEOUT INST.
046040  005767  171316          8$:   TST   $TMP0             ;ANY "ERRORS" DURING TRAP SEQUENCE?
046044  001401                        BEQ   9$
```

```
046046  104000                          EMT                             ;THE WRONG PC OR PSW WERE SAVED
                                                                        ;OR SR0 OR SR2 DID NOT REPORT R/O
                                                                        ;ERROR DURING TIMEOUT - KT TRAP
                                                                        ;SEQUENCE
                                                                        ;FOR TIGHTER SCOPE LOOP
                                                                        ;REPLACE ERROR CALL WITH
                                                                        ;"BR 2$" = 000710
046050  012737  021256  000004  9$:     MOV     #T04,@#4                ;RESTORE ADDRESS OF NORMAL CPU TRAP HANDLER
046056  012737  000340  000006          MOV     #340,@#6                ;RELOAD ERRVEC+2 WITH KERNEL PSW
046064  012737  021276  000250          MOV     #T0250,@#250            ;RESTORE ADDRESS OF NORMAL M.M. TRAP HANDLER
046072  012767  077406  131506          MOV     #77406,UIPDR3           ;REMAP USER PAGE 3 READ/WRITE
046100  004767  002014                  JSR     PC,TON                  ;TURN T-BIT TRAPPING BACK ON
                                ;*******************************************************************************
                                ;*
                                ;*      THIS GROUP OF TESTS WILL TEST ALL THE LOGIC ASSOCIATED WITH
                                ;*      THE "MOVE FROM PREVIOUS" AND MOVE TO PREVIOUS" INSTRUCTIONS.
                                ;*
                                ;*******************************************************************************


                                ;*******************************************************************************
                                ;TEST 413       MOVE FROM PREVIOUS (USER) I-SPACE
                                ;*******************************************************************************
046104                          TS413:
046104  005067  124230          1$:     CLR     KIPAR0                  ;MAP KERNEL PAGE 0 TO 0-4K
046110  012767  000200  124224          MOV     #200,KIPAR1             ;MAP KERNEL PAGE 1 TO 4-8K
046116  012767  000400  124220          MOV     #400,KIPAR2             ;MAP KERNEL PAGE 2 TO 8-12K
046124  012767  000600  124214          MOV     #600,KIPAR3             ;MAP KERNEL PAGE 3 TO 12-16K
046132  012767  001400  124210          MOV     #1400,KIPAR4            ;MAP KERNEL PAGE 4 TO 24-28K
046140  012767  007600  124210          MOV     #7600,KIPAR7            ;MAP KERNEL PAGE 7 TO THE I/O PAGE
046146  012700  077406                  MOV     #77406,R0               ;MAKE ALL KERNEL I-SPACE PAGES RESIDENT
                                                                        ;READ/WRITE, LENGTH 200 BLOCKS
046152  012702  000010                  MOV     #10,R2                  ;SET LOOP COUNTER TO 8
046156  012701  172300                  MOV     #KIPDR0,R1              ;PUT ADDRESS OF FIRST PDR IN R1
046162  010021                  2$:     MOV     R0,(R1)+                ;LOAD PDR WITH 77406
046164  077202                          SOB     R2,2$                   ;LOOP TO 2$ UNTIL ALL PDRS LOADED
046166  012702  000010                  MOV     #10,R2                  ;SET LOOP COUNTER TO 8
046172  012701  177600                  MOV     #UIPDR0,R1              ;PUT ADDRESS OF FIRST PDR IN R1
046176  010021                  3$:     MOV     R0,(R1)+                ;LOAD PDR WITH 77406
046200  077202                          SOB     R2,3$                   ;LOOP TO 3$ UNTIL ALL PDRS LOADED
046202  012767  000000  131430          MOV     #000,UIPAR0             ;MAP USER I PAGE 0 TO 0-4K
046210  012767  000200  131424          MOV     #200,UIPAR1             ;MAP USER I PAGE 1 TO 4-8K
046216  012767  000400  131420          MOV     #400,UIPAR2             ;MAP USER I PAGE 2 TO 8-12K
046224  012767  000600  131414          MOV     #600,UIPAR3             ;MAP USER I PAGE 3 TO 12-16K
046232  012767  007600  131416          MOV     #7600,UIPAR7            ;MAP USER I PAGE 7 TO THE I/O PAGE
046240                          4$:
046240  012767  077406  124042          MOV     #77406,KIPDR4           ;KERNEL I-SPACE PAGE 4 READ/WRITE
046246  012767  001400  124074          MOV     #1400,KIPAR4            ;MAP KERNEL I PAGE 4 TO 24K
046254  012767  001400  131366          MOV     #1400,UIPAR4            ;MAP USER I PAGE 4 TO 24K
046262  012700  036514                  MOV     #36514,R0               ;LOAD DATA PATTERN INTO R0
046266  010037  100000                  MOV     R0,@#100000             ;LOAD DATA PATTERN INTO PHY 140000
046272  012767  046570  131750          MOV     #23$,MMVEC              ;SET M.M. VECTOR TO 23$
046300  105067  124004                  CLRB    KIPDR4                  ;MAKE KERNEL I-SPACE PAGE 4 NON-RESIDENT
                                        ;THE FOLLOWING WILL TEST DSTM=0 MFPI
                                        ;
046304  012767  030340  131464  5$:     MOV     #030340,PSW             ;MAKE PREVIOUS MODE USER
```

```
046312  006506                           6$:    MFPI    USP             ,PUT USER STACK POINTER ON KERNEL
                                                                        ;STACK
046314  022706  001000                          CMP     #KERSTK,KSP     ;WAS SOMETHING PUSHED ON STACK AT 6$
046320  001405                                  BEQ     7$              ;BRANCH IF NOTHING WAS PUSHED
046322  012600                                  MOV     (KSP)+,R0       ;POP KERNEL STACK INTO R0
046324  012701  000600                          MOV     #USESTK,R1      ;EXPECTING TO GET 700 AS USP
046330  020001                                  CMP     R0,R1           ;DID YOU GET THE RIGHT POINTER?
046332  001401                                  BEQ     8$
046334                                   7$:
046334  104000                                  EMT                     ;WRONG THING WAS PUSHED ON STACK
                                                                        ;FOR TIGHTER SCOPE LOOP
                                                                        ;REPLACE ERROR CALL WITH
                                                                        ;"BR 5$" = 000763
046336                                   8$:    ;THE FOLLOWING WILL TEST DSTM=1 MFPI.
046336  012700  036514                          MOV     #36514,R0       ;RELOAD DATA PATTERN IN R0
046342  012767  030340  131426           9$:    MOV     #030340,PSW     ;MAKE PREVIOUS MODE USER
046350  012702  100000                          MOV     #100000,R2      ;LOAD VIRTUAL ADDRESS INTO R2
046354  006512                                  MFPI    (R2)            ;READ FROM PHYSICAL 140000
046356  012601                                  MOV     (KSP)+,R1       ;POP KERNEL STACK INTO R1
046360  020001                                  CMP     R0,R1           ;WAS DATA FETCHED SAME AS STORED
046362  001401                                  BEQ     10$
046364  104000                                  EMT                     ;WRONG DATA WAS FETCHED
                                                                        ;FOR TIGHTER SCOPE LOOP
                                                                        ;REPLACE ERROR CALL WITH
                                                                        ;"BR 9$" = 000766
046366                                   10$:   ;THE FOLLOWING WILL TEST DSTM=2 MFPI.
046366  012767  030340  131402           11$:   MOV     #030340,PSW     ;MAKE PREVIOUS MODE USER
046374  012702  100000                          MOV     #100000,R2      ;LOAD VIRTUAL ADDRESS INTO R2
046400  006522                                  MFPI    (R2)+           ;READ FROM PHYSICAL 140000
046402  012601                                  MOV     (KSP)+,R1       ;POP KERNEL STACK INTO R1
046404  020001                                  CMP     R0,R1           ;WAS DATA FETCHED SAME AS STORED
046406  001401                                  BEQ     12$
046410  104000                                  EMT                     ;WRONG DATA WAS FETCHED
                                                                        ;FOR TIGHTER SCOPE LOOP
                                                                        ;REPLACE ERROR CALL WITH
                                                                        ;"BR 11$" = 000766
046412                                   12$:   ;THE FOLLOWING WILL TEST DSTM=3 MFPI.
046412  012767  030340  131356           13$:   MOV     #030340,PSW     ;MAKE PREVIOUS MODE USER
046420  006537  100000                          MFPI    @#100000        ;READ FROM PHYSICAL 140000
046424  012601                                  MOV     (KSP)+,R1       ;POP KERNEL STACK INTO R1
046426  020001                                  CMP     R0,R1           ;WAS DATA FETCHED SAME AS STORED
046430  001401                                  BEQ     14$
046432  104000                                  EMT                     ;WRONG DATA WAS FETCHED
                                                                        ;FOR TIGHTER SCOPE LOOP
                                                                        ;REPLACE ERROR CALL WITH
                                                                        ;"BR 13$" = 000767
046434                                   14$:   ;THE FOLLOWING WILL TEST DSTM=4 MFPI.
046434  012767  030340  131334           15$:   MOV     #030340,PSW     ;MAKE PREVIOUS MODE USER
046442  012702  100002                          MOV     #100002,R2      ;LOAD VIRTUAL ADDRESS INTO R2
046446  006542                                  MFPI    -(R2)           ;READ FROM PHYSICAL 140000
046450  012601                                  MOV     (KSP)+,R1       ;POP KERNEL STACK INTO R1
046452  020001                                  CMP     R0,R1           ;WAS DATA FETCHED SAME AS STORED
046454  001401                                  BEQ     16$
046456  104000                                  EMT                     ;WRONG DATA WAS FETCHED
                                                                        ;FOR TIGHTER SCOPE LOOP
                                                                        ;REPLACE ERROR CALL WITH
```

```
                                                                ;"BR 15$" = 000766
046460                                  16$:
                                                ;THE FOLLOWING WILL TEST DSTM=5 MFPI.
                                                ;
046460  012767  030340  131310  17$:    MOV     #030340,PSW     ;MAKE PREVIOUS MODE USER
046466  012767  100000  170672          MOV     #100000,$TMP2   ;LOAD TEST LOC. VIRT. ADDR INTO LOC. $TMP2
046474  012702  037370                  MOV     #<$TMP2+2>,R2   ;LOAD ADDR. OF $TMP2+2 INTO R2
046500  006552                          MFPI    @-(R2)          ;READ FROM PHYSICAL 140000
046502  012601                          MOV     (KSP)+,R1       ;POP KERNEL STACK INTO R1
046504  020001                          CMP     R0,R1           ;WAS DATA FETCHED SAME AS STORED
046506  001401                          BEQ     18$
046510  104000                          EMT                     ;WRONG DATA WAS FETCHED
                                                                ;FOR TIGHTER SCOPE LOOP
                                                                ;REPLACE ERROR CALL WITH
                                                                ;"BR 17$" = 000763
046512                                  18$:    ;THE FOLLOWING WILL TEST DSTM=6 MFPI.
                                                ;
046512  012767  030340  131256  19$:    MOV     #030340,PSW     ;MAKE PREVIOUS MODE USER
046520  005002                          CLR     R2              ;MAKE REGISTER 2 A ZERO
046522  006562  100000                  MFPI    100000(R2)      ;READ FROM PHYSICAL 140000
046526  012601                          MOV     (KSP)+,R1       ;POP KERNEL STACK INTO R1
046530  020001                          CMP     R0,R1           ;WAS DATA FETCHED SAME AS STORED
046532  001401                          BEQ     20$
046534  104000                          EMT                     ;WRONG DATA WAS FETCHED
                                                                ;FOR TIGHTER SCOPE LOOP
                                                                ;REPLACE ERROR CALL WITH
                                                                ;"BR 19$" = 000766
046536                                  20$:    ;THE FOLLOWING WILL TEST DSTM=7 MFPI.
                                                ;
046536  012767  030340  131232  21$:    MOV     #030340,PSW     ;MAKE PREVIOUS MODE USER
046544  012767  100000  170614          MOV     #100000,$TMP2   ;LOAD TEST LOC. V.A. INTO $TMP2
046552  012702  037366                  MOV     #$TMP2,R2       ;LOAD ADDRESS OF $TMP2 INTO R2
046556  006572  000000                  MFPI    @0(R2)          ;USE $TMP2 TO FETCH VIRTUAL
                                                                ;ADDRESS OF 140000
046562  012601                          MOV     (KSP)+,R1       ;POP KERNEL STACK INTO R1
046564  020001                          CMP     R0,R1           ;WAS DATA FETCHED SAME AS STORED
046566  001401                          BEQ     22$
046570                                  23$:
046570  104000                          EMT                     ;WRONG DATA WAS FETCHED
                                                                ;FOR TIGHTER SCOPE LOOP
                                                                ;REPLACE ERROR CALL WITH
                                                                ;"BR 21$" = 000762
046572  012767  021276  131450  22$:    MOV     #T0250,MMVEC    ;SET M.M. VECTOR TO NORMAL ROUTINE


                                        ;************************************************************************
                                        ;TEST 414       MOVE TO PREVIOUS (USER) I-SPACE
                                        ;************************************************************************
046600                                  TS414:
046600  012767  077406  123502  1$:     MOV     #77406,KIPDR4   ;KERNEL I-SPACE PAGE 4 READ/WRITE
046606  012767  077406  130774          MOV     #77406,UIPDR4   ;USER I-SPACE PAGE 4 READ/WRITE
046614  012767  001400  123526          MOV     #1400,KIPAR4    ;MAP KERNEL I PAGE 4 TO 24K
046622  012767  001400  131020          MOV     #1400,UIPAR4    ;MAP USER I PAGE 4 TO 24K
046630  012767  047322  131412          MOV     #20$,MMVEC      ;SET M.M. VECTOR TO 20$
                                                ;THE FOLLOWING WILL TEST DSTM=0 MTPI
                                                ;
```

```
046636  012767  030340  131132  2$:     MOV     #030340,PSW      ;MAKE PREVIOUS MODE USER
046644  012746  007777                  MOV     #7777,-(KSP)     ;PUSH DATA ON KERNEL STACK
046650  006606                          MTPI    USP              ;LOAD USER STACK POINTER
046652  006506                          MFPI    USP              ;READ USER STACK POINTER
046654  012601                          MOV     (KSP)+,R1        ;POP KERNEL STACK INTO R1
046656  022701  007777                  CMP     #7777,R1         ;WAS USER STACK POINTER CHANGED
046662  001401                          BEQ     3$
046664  104000                          EMT                      ;USER STACK POINTER NOT CHANGED
                                                                 ;FOR TIGHTER SCOPE LOOP
                                                                 ;REPLACE ERROR CALL WITH
                                                                 ;"BR 2$" = 000764
046666  012767  030340  131102  3$:     MOV     #030340,PSW      ;MAKE PREVIOUS MODE USER
046674  012746  000600                  MOV     #USESTK,-(KSP)   ;GET READY TO RESTORE USER S. POINT
046700  006606                          MTPI    USP              ;RESTORE USER STACK POINTER
046702                          4$:     ;THIS WILL TEST DSTM = 1 MTPI.
046702  012702  100000                  MOV     #100000,R2       ;LOAD VIRTUAL ADDRESS INTO R2
046706  012700  125252                  MOV     #125252,R0       ;LOAD TEST DATA INTO R0
046712  010046                  5$:     MOV     R0,-(KSP)        ;PUSH TEST DATA ON KERNEL STACK
046714  105067  123370                  CLRB    KIPDR4           ;MAKE KERNEL I PAGE 4 NON-RESIDENT
046720  006612                          MTPI    (R2)             ;LOAD TEST DATA INTO PHYSICAL 140000
046722  112767  000006  123360          MOVB    #006,KIPDR4      ;MAKE KERNEL PAGE 4 RESIDENT
046730  011201                          MOV     (R2),R1          ;READ FROM ADDRESS 140000
046732  020001                          CMP     R0,R1            ;SEE IF DATA WAS STORED AT CORRECT PLACE
046734  001401                          BEQ     6$
046736  104000                          EMT                      ;INCORRECT STORE
                                                                 ;FOR TIGHTER SCOPE LOOP
                                                                 ;REPLACE ERROR CALL WITH
                                                                 ;"BR 5$" = 000765
046740                          6$:     ;THE FOLLOWING WILL TEST DSTM=2 MTPI.
                                        ;
046740  012767  030340  131030          MOV     #030340,PSW      ;MAKE PREVIOUS MODE USER
046746  012700  125252                  MOV     #125252,R0       ;LOAD TEST DATA INTO R0
046752  012702  100000                  MOV     #100000,R2       ;LOAD VIRTUAL ADDRESS INTO R2
046756  010046                  8$:     MOV     R0,-(KSP)        ;PUSH TEST DATA ON KERNEL STACK
046760  105067  123324                  CLRB    KIPDR4           ;MAKE KERNEL PAGE 4 NON-RESIDENT
046764  006612                          MTPI    (R2)             ;LOAD TEST DATA INTO PHYSICAL 140000
046766  112767  000006  123314          MOVB    #006,KIPDR4      ;MAKE KERNEL PAGE 4 RESISENT
046774  013701  100000                  MOV     @#100000,R1      ;READ FROM ADDRESS 140000
047000  020001                          CMP     R0,R1            ;SEE IF DATA WAS STORED CORRECTLY
047002  001401                          BEQ     9$
047004  104000                          EMT                      ;INCORRECT STORE
                                                                 ;FOR TIGHTER SCOPE LOOP
                                                                 ;REPLACE ERROR CALL WITH
                                                                 ;"BR 8$" = 000764
047006                          9$:     ;THIS WILL TEST DSTM = 3 MTPI.
047006  012767  030340  130762          MOV     #030340,PSW      ;MAKE PREVIOUS MODE USER
047014  012700  052525                  MOV     #52525,R0        ;LOAD TEST DATA INTO R0
047020  010046                  10$:    MOV     R0,-(KSP)        ;PUSH TEST DATA ON KERNEL STACK
047022  105067  123262                  CLRB    KIPDR4           ;MAKE KERNEL I PAGE 4 NON-RESIDENT
047026  006637  100000                  MTPI    @#100000         ;LOAD TEST DATA INTO PHYSICAL 140000
047032  112767  000006  123250          MOVB    #006,KIPDR4      ;MAKE KERNEL PAGE 4 RESIDENT
047040  013701  100000                  MOV     @#100000,R1      ;READ FROM ADDRESS 140000
047044  020001                          CMP     R0,R1            ;SEE IF DATA WAS STORED CORRECTLY
047046  001531                          BEQ     TS415
047050  104000                          EMT                      ;
047052  001401                          BEQ     11$
```

```
047054  104000                          EMT                             ;INCORRECT STORE
                                                                        ;FOR TIGHTER SCOPE LOOP
                                                                        ;REPLACE ERROR CALL WITH
                                                                        ;"BR 10$" = 000763
047056                          11$:    ;THIS WILL TEST DSTM = 4 MTPI.
047056  012767  030340  130712          MOV     #030340,PSW             ;MAKE PREVIOUS MODE USER
047064  012700  125252                  MOV     #125252,R0              ;LOAD TEST DATA INTO R0
047070  010046                  12$:    MOV     R0,-(KSP)               ;PUSH TEST DATA ON KERNEL STACK
047072  012702  100002                  MOV     #100002,R2              ;LOAD VIRTUAL ADDRESS INTO R2
047076  105067  123206                  CLRB    KIPDR4                  ;MAKE KERNEL I PAGE 4 NON-RESIDENT
047102  006642                          MTPI    -(R2)                   ;LOAD TEST DATA INTO PHYSICAL 140000
047104  112767  000006  123176          MOVB    #006,KIPDR4             ;MAKE KERNEL PAGE 4 RESIDENT
047112  013701  100000                  MOV     @#100000,R1             ;READ FROM ADDRESS 140000
047116  020001                          CMP     R0,R1                   ;SEE IF DATA WAS STORED CORRECTLY
047120  001401                          BEQ     13$
047122  104000                          EMT                             ;INCORRECT STORE
                                                                        ;FOR TIGHTER SCOPE LOOP
                                                                        ;REPLACE ERROR CALL WITH
                                                                        ;"BR 12$" = 000762
047124                          13$:    ;THE FOLLOWING WILL TEST DSTM=5 MTPI.
                                        ;
047124  012767  030340  130644          MOV     #030340,PSW             ;MAKE PREVIOUS MODE USER
047132  012700  052525                  MOV     #52525,R0               ;LOAD TEST DATA INTO R0
047136  012702  037370                  MOV     #<$TMP2+2>,R2           ;LOAD ADDR. OF LOC. $TMP2+2 INTO R2
047142  012767  100000  170216          MOV     #100000,$TMP2           ;LOAD VIRT. ADDR. OF TEST LOC. INTO $TMP2
047150  010046                  14$:    MOV     R0,-(KSP)               ;PUSH TEST DATA ON KERNEL STACK
047152  105067  123132                  CLRB    KIPDR4                  ;MAKE KERNEL PAGE 4 NON-RESIDENT
047156  006652                          MTPI    @-(R2)                  ;LOAD TEST DATA INTO PHYSICAL 140000
047160  112767  000006  123122          MOVB    #006,KIPDR4             ;MAKE KERNEL PAGE 4 RESIDENT
047166  013701  100000                  MOV     @#100000,R1             ;READ FROM ADDRESS 140000
047172  020001                          CMP     R0,R1                   ;SEE IF DATA WAS STORED CORRECTLY
047174  001401                          BEQ     15$
047176  104000                          EMT                             ;INCORRECT STORE
                                                                        ;FOR TIGHTER SCOPE LOOP
                                                                        ;REPLACE ERROR CALL WITH
                                                                        ;"BR 14$" = 000764
047200                          15$:    ;THIS WILL TEST DSTM = 6 MTPI.
                                        ;
047200  012767  030340  130570          MOV     #030340,PSW             ;MAKE PREVIOUS MODE USER
047206  012700  052525                  MOV     #52525,R0               ;LOAD TEST DATA INTO R0
047212  005002                          CLR     R2                      ;MAKE REGISTER 2 ZERO
047214  010046                  16$:    MOV     R0,-(KSP)               ;PUSH TEST DATA ON KERNEL STACK
047216  105067  123066                  CLRB    KIPDR4                  ;MAKE KERNEL I PAGE 4 NON-RESIDENT
047222  006662  100000                  MTPI    100000(R2)              ;LOAD TEST DATA INTO PHYSICAL 140000
047226  112767  000006  123054          MOVB    #006,KIPDR4             ;MAKE KERNEL PAGE 4 RESIDENT
047234  013701  100000                  MOV     @#100000,R1             ;READ FROM ADDRESS 140000
047240  020001                          CMP     R0,R1                   ;SEE IF DATA WAS STORED CORRECTLY
047242  001401                          BEQ     17$
047244  104000                          EMT                             ;INCORRECT STORE
                                                                        ;FOR TIGHTER SCOPE LOOP
                                                                        ;REPLACE ERROR CALL WITH
                                                                        ;"BR 16$" = 000763
047246                          17$:    ;THE FOLLOWING WILL TEST DSTM=7 MTPI.
                                        ;
047246  012767  030340  130522          MOV     #030340,PSW             ;MAKE PREVIOUS MODE USER
047254  012700  125252                  MOV     #125252,R0              ;LOAD TEST DATA INTO R0
```

```
047260 012767 100000 170100          MOV     #100000,$TMP2    ;LOAD VIRT. ADDR. OF TEST LOCATION
                                                              ;INTO LOCATION $TMP2
047266 012702 037366                 MOV     #$TMP2,R2        ;LOAD ADDRESS OF $TMP2 INTO R2
047272 010046                18$:    MOV     R0,-(KSP)        ;PUSH TEST DATA ON KERNEL STACK
047274 105067 123010                 CLRB    KIPDR4           ;MAKE KERNEL PAGE 4 NON-RESIDENT
047300 006672 000000                 MTPI    @0(R2)           ;LOAD TEST DATA INTO PHYSICAL 140000
047304 112767 000006 122776          MOVB    #006,KIPDR4      ;MAKE KERNEL PAGE 4 RESIDENT
047312 013701 100000                 MOV     @#100000,R1      ;READ FROM ADDRESS 140000
047316 020001                        CMP     R0,R1            ;SEE IF DATA WAS STORED CORRECTLY
047320 001401                        BEQ     19$
047322                       20$:
047322 104000                        EMT                      ;INCORRECT STORE
                                                              ;FOR TIGHTER SCOPE LOOP
                                                              ;REPLACE ERROR CALL WITH
                                                              ;"BR 18$" = 000763
047324 012767 021276 130716  19$:    MOV     #T0250,MMVEC     ;RESTORE M.M. VECTOR TO NORMAL ROUTINE



                             ;*****************************************************************************
                             ;TEST 415       MOVE FROM PREVIOUS (KERNEL) I-SPACE TO USER MODE
                             ;*****************************************************************************
047332                       TS415:
047332 012700 077406         1$:     MOV     #77406,R0        ;MAKE ALL USER I-SPACE PAGES RESIDENT
                                                              ;READ/WRITE, LENGTH 200 BLOCKS
047336 012702 000010                 MOV     #10,R2           ;SET LOOP COUNTER TO 8
047342 012701 177600                 MOV     #UIPDR0,R1       ;LOAD ADDRESS OF FIRST PDR IN R1
047346 010021                2$:     MOV     R0,(R1)+         ;LOAD PDR WITH 77406
047350 077202                        SOB     R2,2$            ;LOOP UNTIL 8 USER PDRS LOADED
047352 012767 140340 130416  3$:     MOV     #140340,PSW      ;GO TO USER MODE FOR THIS TEST
047360 012767 077406 122722          MOV     #77406,KIPDR4    ;KERNEL I-SPACE PAGE 4 READ/WRITE
047366 012767 001400 122754          MOV     #1400,KIPAR4     ;MAP KERNEL I PAGE 4 TO 24K
047374 012767 001400 130246          MOV     #1400,UIPAR4     ;MAP USER I PAGE 4 TO 24K
047402 012700 036514                 MOV     #36514,R0        ;LOAD DATA PATTERN INTO R0
047406 010037 100000                 MOV     R0,@#100000      ;LOAD DATA PATTERN INTO PHY 140000
047412 012702 100000                 MOV     #100000,R2       ;LOAD VIRTUAL ADDRESS INTO R2
                                     ;THE FOLLOWING WILL TEST DSTM=0 MFPI
                                     ;
047416 012767 047714 130624          MOV     #21$,MMVEC       ;SET M.M. VECTOR TO 21$
047424 105067 130160                 CLRB    UIPDR4           ;MAKE USER I-SPACE PAGE 4 NON-RESIDENT
047430 012767 140340 130340          MOV     #140340,PSW      ;MAKE PREVIOUS MODE KERNEL PRESENT USER
047436 006506                4$:     MFPI    KSP              ;PUT KERNEL STACK POINTER ON USER STACK
047440 022706 000600                 CMP     #USESTK,USP      ;WAS SOMETHING PUSHED ON STACK AT 1$
047444 001405                        BEQ     5$               ;BRANCH IF NOTHING WAS PUSHED
047446 012600                        MOV     (USP)+,R0        ;POP USER STACK INTO R0
047450 012701 001000                 MOV     #KERSTK,R1       ;EXPECTING 1100 AS KSP
047454 020001                        CMP     R0,R1            ;DID YOU GET THE RIGHT POINTER?
047456 001401                        BEQ     6$
047460                       5$:
047460 104000                        EMT                      ;WRONG THING WAS PUSHED ON STACK
                                                              ;FOR TIGHTER SCOPE LOOP
                                                              ;REPLACE ERROR CALL WITH
                                                              ;"BR 4$" = 000766
047462                       6$:     ;THE FOLLOWING WILL TEST DSTM=1 MFPI.
047462 012767 140340 130306  7$:     MOV     #140340,PSW      ;MAKE PREVIOUS MODE KERNEL PRESENT USER
047470 012700 036514                 MOV     #36514,R0        ;LOAD DATA EXPECTED INTO R0
```

```
047474  012702  100000                  MOV     #100000,R2      ;LOAD VIRTUAL ADDRESS INTO R2
047500  006512                          MFPI    (R2)            ;READ FROM PHYSICAL 140000
047502  012601                          MOV     (USP)+,R1       ;POP USER STACK INTO R1
047504  020001                          CMP     R0,R1           ;WAS DATA FETCHED SAME AS STORED
047506  001401                          BEQ     9$
047510  104000                          EMT                     ;WRONG DATA WAS FETCHED
                                                                ;FOR TIGHTER SCOPE LOOP
                                                                ;REPLACE ERROR CALL WITH
                                                                ;"BR 7$" = 000764
                                        ;THE FOLLOWING WILL TEST DSM=2 MFPI.
047512  012767  140340  130256  9$:     MOV     #140340,PSW     ;MAKE PREVIOUS MODE KERNEL PRESENT USER
047520  012702  100000                  MOV     #100000,R2      ;LOAD VIRTUAL ADDRESS INTO R2
047524  006522                          MFPI    (R2)+           ;READ FROM PHYSICAL 140000
047526  012601                          MOV     (USP)+,R1       ;POP USER STACK INTO R1
047530  020001                          CMP     R0,R1           ;WAS DATA FETCHED SAME AS STORED
047532  001401                          BEQ     11$
047534  104000                          EMT                     ;WRONG DATA WAS FETCHED
                                                                ;FOR TIGHTER SCOPE LOOP
                                                                ;REPLACE ERROR CALL WITH
                                                                ;"BR 9$" = 000766
                                        ;THE FOLLOWING WILL TEST DSTM=3 MFPI.
047536  012767  140340  130232  11$:    MOV     #140340,PSW     ;MAKE PREVIOUS MODE KERNEL PRESENT USER
047544  006537  100000                  MFPI    @#100000        ;READ FROM PHYSICAL 140000
047550  012601                          MOV     (USP)+,R1       ;POP USER STACK INTO R1
047552  020001                          CMP     R0,R1           ;WAS DATA FETCHED SAME AS STORED
047554  001401                          BEQ     13$
047556  104000                          EMT                     ;WRONG DATA WAS FETCHED
                                                                ;FOR TIGHTER SCOPE LOOP
                                                                ;REPLACE ERROR CALL WITH
                                                                ;"BR 11$" = 000767
                                        ;THE FOLLOWING WILL TEST DSTM=4 MFPI.
047560  012767  140340  130210  13$:    MOV     #140340,PSW     ;MAKE PREVIOUS MODE DERNEL PRESENT USER
047566  012702  100002                  MOV     #100002,R2      ;LOAD VIRTUAL ADDRESS INTO R2
047572  006542                          MFPI    -(R2)           ;READ FROM PHYSICAL 140000
047574  012601                          MOV     (USP)+,R1       ;POP USER STACK INTO R1
047576  020001                          CMP     R0,R1           ;WAS DATA FETCHED SAME AS STORED
047600  001401                          BEQ     15$
047602  104000                          EMT                     ;WRONG DATA WAS FETCHED
                                                                ;FOR TIGHTER SCOPE LOOP
                                                                ;REPLACE ERROR CALL WITH
                                                                ;"BR 13$" = 000766
                                        ;THE FOLLOWING WILL TEST DSTM=5 MFPI.
                                        ;
047604  012767  140340  130164  15$:    MOV     #140340,PSW     ;MAKE PREVIOUS MODE KERNEL PRESENT USER
047612  012767  100000  167546          MOV     #100000,$TMP2   ;LOAD TEST LOC. VIRT. ADDR INTO LOC. $TMP2
047620  012702  037370                  MOV     #<$TMP2+2>,R2   ;LOAD ADDRESS OF $TMP2+2 INTO R2
047624  006552                          MFPI    @-(R2)          ;READ FROM PHYSICAL 140000
047626  012601                          MOV     (USP)+,R1       ;POP USER STACK INTO R1
047630  020001                          CMP     R0,R1           ;WAS DATA FETCHED SAME AS STORED
047632  001401                          BEQ     17$
047634  104000                          EMT                     ;WRONG DATA WAS FETCHED
                                                                ;FOR TIGHTER SCOPE LOOP
                                                                ;REPLACE ERROR CALL WITH
                                                                ;"BR 15$" = 000763
                                        ;THE FOLLOWING WILL TEST DSTM=6 MFPI.
                                        ;
```

```
047636  012767  140340  130132  17$:    MOV     #140340,PSW     ;MAKE PREVIOUS MODE KERNEL PRESENT USER
047644  005002                          CLR     R2              ;MAKE REGISTER 2 A ZERO
047646  006562  100000                  MFPI    100000(R2)      ;READ FROM PHYSICAL 140000
047652  012601                          MOV     (USP)+,R1       ;POP USER STACK INTO R1
047654  020001                          CMP     R0,R1           ;WAS DATA FETCHED SAME AS STORED
047656  001401                          BEQ     19$
047660  104000                          EMT                     ;WRONG DATA WAS FETCHED
                                                                ;FOR TIGHTER SCOPE LOOP
                                                                ;REPLACE ERROR CALL WITH
                                                                ;"BR 17$" = 000766
                                        ;THE FOLLOWING WILL TEST DSTM=7 MFPI.
                                        ;
047662  012767  140340  130106  19$:    MOV     #140340,PSW     ;MAKE PREVIOUS MODE KERNEL PRESENT USER
047670  012767  100000  167470          MOV     #100000,$TMP2   ;LOAD TEST LOC. VIRT. ADDR. INTO $TMP2
047676  012702  037366                  MOV     #$TMP2,R2       ;LOAD ADDRESS OF $TMP2 INTO R2
047702  006572  000000                  MFPI    @0(R2)          ;READ FROM PHYSICAL 140000
047706  012601                          MOV     (USP)+,R1       ;POP USER STACK INTO R1
047710  020001                          CMP     R0,R1           ;WAS DATA FETCHED SAME AS STORED
047712  001401                          BEQ     20$
047714                          21$:
047714  104000                          EMT                     ;WRONG DATA WAS FETCHED
                                                                ;FOR TIGHTER SCOPE LOOP
                                                                ;REPLACE ERROR CALL WITH
                                                                ;"BR 19$" = 000762
047716  012767  021276  130324  20$:    MOV     #T0250,MMVEC    ;SET M.M. VECTOR TO NORMAL ROUTINE
047724  012767  000340  130044          MOV     #00340,PSW      ;GO BACK TO KERNEL MODE, PREVIOUS KERNEL

                                ;*******************************************************************************
                                ;TEST 416       MOVE FROM/TO D-SPACE = MOVE FROM/TO I-SPACE
                                ;*******************************************************************************
047732                          TS416:
047732  012767  030340  130036  1$:     MOV     #030340,PSW     ;MAKE PREVIOUS MODE=USER,CURRENT=KERNEL
047740  106506                          MFPD    USP             ;MFPD SHOULD ACT LIKE MFPI PUTTING
                                                                ;USER STACK POINTER ON THE KERNEL STACK
047742  022706  001000                  CMP     #KERSTK,KSP     ;WAS SOMETHING PUSHED ON KERNEL STACK?
047746  001405                          BEQ     2$              ;BRANCH IF NO
047750  012600                          MOV     (KSP)+,R0       ;POP KERNEL STACK INTO R0
047752  012701  000600                  MOV     #USESTK,R1      ;EXPECTING TO GET 700 AS USP
047756  020001                          CMP     R0,R1           ;DID GET RIGHT POINTER VALUE?
047760  001401                          BEQ     4$
047762                          2$:
047762  104000                          EMT                     ;WRONG THING WAS PUSHED ON STACK
                                                                ;FOR TIGHTER SCOPE LOOP
                                                                ;REPLACE ERROR CALL WITH
                                                                ;"BR 1$" = 000763
047764  012746  007777          4$:     MOV     #7777,-(KSP)    ;PUSH DATA ON KERNEL STACK
047770  106606                          MTPD    USP             ;LOAD THE USER STACK POINTER
047772  106506                          MFPD    USP             ;READ USER STACK POINTER
047774  012601                          MOV     (KSP)+,R1       ;POP KERNEL STACK INTO R1
047776  022701  007777                  CMP     #7777,R1        ;WAS USER STACK POINTER CHANGED?
050002  001401                          BEQ     5$
050004  104000                          EMT                     ;USER STACK POINTER NOT CHANGED
                                                                ;FOR TIGHTER SCOPE LOOP
                                                                ;REPLACE ERROR CALL WITH
                                                                ;"BR 4$" = 000767
050006  012746  000600          5$:     MOV     #USESTK,-(KSP)  ;GET READY TO RESTORE USER STK. PTR.
```

```
050012  106606                          MTPD    USP                 ;RESTORE USER STACK POINTER

                                ;*******************************************************************************
                                ;TEST 417       MOVE FROM PREVIOUS I=SPACE (PREVIOUS=CURRENT=KERNEL)
                                ;*******************************************************************************
050014                          TS417:
050014  005037  177776          1$:     CLR     @#PSW               ;SET PREVIOUS = CURRENT = KERNEL
050020  012700  001000                  MOV     #KERSTK,R0          ;SETUP VALUE FOR STACK POINTER
050024  010006                          MOV     R0,KSP              ;LOAD STACK POINTER
050026  006506                          MFPI    KSP                 ;THE VALUE "STACK" SHOULD BE PUSHED
                                                                    ;BEFORE BEING DECREMENTED
050030  011601                          MOV     (KSP),R1            ;READ DATA WHICH WAS PUSHED
050032  020001                          CMP     R0,R1               ;WAS THE ORIGINAL VALUE OF THE
                                                                    ;STACK POINTER PUSHED?
050034  001401                          BEQ     2$
050036  104000                          EMT                         ;MFPI FETCHED WRONG DATA
                                                                    ;FOR TIGHTER SCOPE LOOP
                                                                    ;REPLACE ERROR CALL WITH
                                                                    ;"BR 1$" = 000766
050040  005740                  2$:     TST     -(R0)               ;SETUP EXPECTED STACK POINTER VALUE
050042  020600                          CMP     KSP,R0              ;WAS THE STACK POINTER DECREMENTED?
050044  001401                          BEQ     3$
050046  104000                          EMT                         ;STACK NOT PUSHED BY THE MFPI
                                                                    ;FOR TIGHTER SCOPE LOOP
                                                                    ;REPLACE ERROR CALL WITH
                                                                    ;"BR 1$" = 000762
050050  012706  001000          3$:     MOV     #KERSTK,KSP         ;RESTORE STACK POINTER
050054  005067  127512                  CLR     SR0                 ;TURN OFF MEMORY MANAGEMENT UNIT
050060  000167  000436                  JMP     FPSTRT              ;GET OVER SUBROUTINES TO FLOATING POINT TE TS
```

```
                                      .SBTTL  *****   SUBROUTINES USED BY THIS PROGRAM   *****

                                      .SBTTL  TURN OFF T-BIT AND SAVE CURRENT PSW
                                      ;*********************************************************************
                                      ;*
                                      ;*      THIS SUBROUTINE IS USED TO TURN OFF THE TRACE TRAP BIT IN THE PSW
                                      ;*      IF IT IS ON.  THE PROCESSOR STATUS IS SAVED IN "TBITPS" SO THAT
                                      ;*      THE PSW CAN BE RESTORED TO ITS PREVIOUS CONDITION WHEN CONDITIONS
                                      ;*      WARRANT T-BIT TRAPPING.
                                      ;*
                                      ;*********************************************************************
050064  036727  127706  000020        TOFF:   BIT     PSW,#TBIT          ;IS THE T-BIT SET IN THE PSW?
050072  001411                                BEQ     1$                 ;EXIT IF NO
050074  016746  127676                        MOV     PSW,-(SP)          ;PUSH PRESENT PSW ON THE STACK
050100  011667  167254                        MOV     (SP),TBITPS        ;ALSO SAVE IT IN "TBITPS" FOR
                                                                         ;RESTORING LATER
050104  042716  000020                        BIC     #TBIT,(SP)         ;CLEAR THE T-BIT (BIT 4) IN THE PSW
050110  012746  050116                        MOV     #1$,-(SP)          ;PUSH PC OF "RTS" ON STACK
050114  000006                                RTT                        ;"RETURN" TO 1$ WITH T-BIT OFF
050116  000207                        1$:     RTS     PC                 ;RETURN TO PROGRAM

                                      .SBTTL  TURN ON T-BIT AND RESTORE PREVIOUS PSW
                                      ;*********************************************************************
                                      ;*
                                      ;*      THIS SUBROUTINE IS USED TO RESTORE THE PROCESSOR STATUS TO ITS
                                      ;*      PREVIOUS CONDITION BY RESTORING THE "T-BIT PSW" SAVED BY THE
                                      ;*      "TOFF" SUBROUTINE IN THE "TBITPS" LOCATION.
                                      ;*
                                      ;*********************************************************************
050120  036727  167234  000020        TON:    BIT     TBITPS,#TBIT       ;WAS T-BIT ON IN THE PREVIOUS PSW?
050126  001410                                BEQ     1$                 ;EXIT IF NO
050130  016746  167224                        MOV     TBITPS,-(SP)       ;PUSH PREVIOUS PSW ON THE STACK
050134  012767  000340  167216                MOV     #340,TBITPS        ;RESET THE "TBITPS" LOCATION
050142  012746  050150                        MOV     #1$,-(SP)          ;PUSH PC OF "RTS" ON STACK
050146  000006                                RTT                        ;"RETURN" TO 1$ WITH T-BIT RESTORED
050150  000207                        1$:     RTS     PC                 ;RETURN TO PROGRAM

                                      .SBTTL  SET ALL WRITEABLE BITS IN ALL PAR/PDR'S
                                      ;*********************************************************************
                                      ;*
                                      ;*      THIS SUBROUTINE IS USED BY THE PAR/PDR DUAL ADDRESSING TEST
                                      ;*      TO SET ALL WRITEABLE BITS IN ALL KERNEL AND USE PAR'S AND
                                      ;*      PDR'S TO A 1.  THE "INITIAL STATE" OF HAVING ALL BITS=1 IS
                                      ;*      USED TO SEE THAT ONLY ONE REGISTER IS CLEARED IN RESPONSE TO
                                      ;*      A SINGLE PAR OR PDR ADDRESS.
                                      ;*
                                      ;*********************************************************************

050152  012702  000010                SETREG: MOV     #10,R2             ;LOAD LOOP COUNTER WITH AN 8
050156  012701  172300                        MOV     #KIPDR0,R1         ;LOAD ADDRESS OF FIRST PDR INTO R1
050162  012721  177777                1$:     MOV     #-1,(R1)+          ;SET BITS IN KERNEL PDR TO 1
050166  077203                                SOB     R2,1$              ;LOOP TO 1$ UNTIL ALL KERNEL PDR'S LOADED
050170  012702  000010                        MOV     #10,R2             ;LOAD LOOP COUNTER WITH AN 8
050174  012701  172340                        MOV     #KIPAR0,R1         ;LOAD ADDRESS OF FIRST PAR INTO R1
050200  012721  177777                2$:     MOV     #-1,(R1)+          ;SET BITS IN A KERNEL PAR TO 1
050204  077203                                SOB     R2,2$              ;LOOP TO 2$ UNTIL ALL KERNEL PAR'S LOADED
```

```
050206  012702  000010              MOV     #10,R2          ;LOAD LOOP COUNTER WITH AN 8
050212  012701  177600              MOV     #UIPDRO,R1      ;LOAD ADDRESS OF FIRST PDR INTO R1
050216  012721  177777      3$:     MOV     #-1,(R1)+       ;SET BITS IN A USER PDR TO 1
050222  077203                      SOB     R2,3$           ;LOOP TO 3$ UNTIL ALL USER PDR'S LOADED
050224  012702  000010              MOV     #10,R2          ;LOAD LOOP COUNTER WITH AN 8
050230  012701  177640              MOV     #UIPARO,R1      ;LOAD ADDRESS OF FIRST PAR INTO R1
050234  012721  177777      4$:     MOV     #-1,(R1)+       ;SET BITS IN A USER PAR TO 1
050240  077203                      SOB     R2,4$           ;LOOP TO 4$ UNTIL ALL USER PAR'S LOADED
050242  000207                      RTS     PC              ;RETURN TO TEST

                            .SBTTL  READ & COMPARE KERNEL & USER PAR/PDR'S
                            ;*****************************************************************************
                            ;*
                            ;*      THIS SUBROUTINE IS USED BY PAR/PDR DUAL ADDRESSING TEST TO
                            ;*      READ ALL THE PAR'S AND PDR'S TO SEE THAT ONLY ONE REGISTER
                            ;*      WAS CLEARED IN RESPONSE TO A SINGLE PAR OR PDR ADDRESS.
                            ;*      ANY FAILURES FOUND BY THE PAR/PDR DUAL ADDRESSING TEST WILL
                            ;*      BE REPORTED BY THIS SUBROUTINE.
                            ;*
                            ;*****************************************************************************
050244                      CMPREG:
050244  012701  172300              MOV     #KIPDRO,R1      ;LOAD ADDRESS OF FIRST KERNEL PDR IN R1
050250  012704  000010              MOV     #10,R4          ;LOAD LOOP COUNTER WITH AN 8
050254  012705  077416              MOV     #77416,R5       ;PUT EXPECTED PDR CONTENTS IN R5
050260  021105              1$:     CMP     (R1),R5         ;ARE ALL WRITEABLE BITS SET AS EXPECTED?
050262  001403                      BEQ     2$              ;BRANCH IF YES
050264  020100                      CMP     R1,R0           ;WAS IT THE REG. THAT WAS CLEARED?
050266  001401                      BEQ     2$
050270  104000                      EMT                     ;A PDR WAS EFFECTED BY CLEARING A DIFFERENT PAR/PRD
                                                            ;FOR TIGHTER SCOPE LOOP
                                                            ;REPLACE ERROR CALL WITH
                                                            ;AN 'RTS PC'' = 000207
050272  062701  000002      2$:     ADD     #2,R1           ;FORM NEXT ADDRESS
050276  077410                      SOB     R4,1$           ;LOOP TO 1$ UNTIL ALL KERNEL PDR'S CHECKED
050300  012701  172340              MOV     #KIPARO,R1      ;LOAD ADDRESS OF FIRST KERNEL PAR IN R1
050304  012704  000010              MOV     #10,R4          ;LOAD LOOP COUNTER WITH AN 8
                                                            ;****F11 CHANGE**** FROM #7777 TO #177777
050310  012705  177777              MOV     #177777,R5      ;PUT EXPECTED PAR CONTENTS IN R5
050314  021105              3$:     CMP     (R1),R5         ;ARE ALL WRITEABLE BITS SET AS EXPECTED?
050316  001403                      BEQ     4$              ;BRANCH IF YES
050320  020100                      CMP     R1,R0           ;WAS IT THE REG. THAT WAS CLEARED?
050322  001401                      BEQ     4$
050324  104000                      EMT                     ;A PAR WAS EFFECTED BY CLEARING A DIFFENENT PAR/PDR
                                                            ;FOR TIGHTER SCOPE LOOP
                                                            ;REPLACE ERROR CALL WITH
                                                            ;AN 'RTS PC'' = 000207
050326  062701  000002      4$:     ADD     #2,R1           ;FORM NEXT ADDRESS
050332  077410                      SOB     R4,3$           ;LOOP TO 3$ UNTIL ALL KERNEL PAR'S CHECKED
050334  012701  177600              MOV     #UIPDRO,R1      ;LOAD ADDRESS OF FIRST USER PDR IN R1
050340  012704  000010              MOV     #10,R4          ;LOAD LOOP COUNTER WITH AN 8
050344  012705  077416              MOV     #77416,R5       ;PUT EXPECTED PDR CONTENTS IN R5
050350  021105              5$:     CMP     (R1),R5         ;ARE ALL WRITEABLE BITS SET AS EXPECTED?
050352  001403                      BEQ     6$              ;BRANCH IF YES
050354  020100                      CMP     R1,R0           ;WAS IT THE REG. THAT WAS CLEARED?
050356  001401                      BEQ     6$
050360  104000                      EMT                     ;A PDR WAS EFFECTED BY CLEARING A DIFFERENT PAR/PDR
```

```
                                                                ;FOR TIGHTER SCOPE LOOP
                                                                ;REPLACE ERROR CALL WITH
                                                                ;AN "RTS PC" = 000207
050362  062701  000002          6$:     ADD     #2,R1           ;FORM NEXT ADDRESS
050366  077410                          SOB     R4,5$           ;LOOP TO 5$ UNTIL ALL USER PDR'S CHECKED
050370  012701  177640                  MOV     #UIPAR0,R1      ;LOAD ADDRESS OF FIRST USER PAR IN R1
050374  012704  000010                  MOV     #10,R4          ;LOAD LOOP COUNTER WITH AN 8
                                                                ;****F11 CHANGE**** FROM #7777 TO #177777
050400  012705  177777                  MOV     #177777,R5      ;PUT EXPECTED PAR CONTENTS IN R5
050404  021105                  7$:     CMP     (R1),R5         ;ARE ALL WRITEABLE BITS SET AS EXPECTED?
050406  001403                          BEQ     8$              ;BRANCH IF YES
050410  020100                          CMP     R1,R0           ;WAS IT THE REG. THAT WAS CLEARED?
050412  001401                          BEQ     8$
050414  104000                          EMT                     ;A PAR WAS EFFECTED BY CLEARING A DIFFERENT PAR/PDR
                                                                ;FOR TIGHTER SCOPE LOOP
                                                                ;REPLACE ERROR CALL WITH
                                                                ;AN "RTS PC" = 000207
050416  062701  000002          8$:     ADD     #2,R1           ;FORM NEXT ADDRESS
050422  077410                          SOB     R4,7$           ;LOOP TO 7$ UNTIL ALL USER PAR'S CHECKED
050424  000207                          RTS     PC              ;RETURN TO TEST

                                .SBTTL  INHIBIT 'RESETS' WHILE UNDER APT
                                ;*****************************************************************************
                                ;*
                                ;*      THIS SUBROUTINE CONTROLS THE USAGE OF RESET INST'S WHILE
                                ;*      RUNNING UNDER APT. RESETS ARE ALLOWED DURING THE FIRST
                                ;*      PASS OF THE DIAGNOSTIC.
                                ;*
050426  126727  130366  000001  CHKAPT: CMPB    $ENV,#1         ;ARE WE RUNNING UNDER APT?
050434  001003                          BNE     1$              ;NO BRANCH
050436  005767  130344                  TST     $PASS           ;IS THIS THE FIRST PASS?
050442  001002                          BNE     RETA            ;NO BRANCH
050444  062705  000002          1$:     ADD     #2,R5           ;BUMP RETURN ADDRESS FOR NORMAL TESTING
050450  000205                  RETA:   RTS     R5              ;RETURN


                                .SBTTL  ERROR ROUTINE FOR MEMORY MANAGEMENT TEST
                                ;*****************************************************************************
                                ;*      THIS IS THE ONLY ERROR REPORT FOR ALL THE MMU TESTS
                                ;*****************************************************************************

050452  012737  050504  000004  ERROR3: MOV     #1$,@#4         ;SET UP THE TIMEOUT VECTOR
050460  012737  000340  000006          MOV     #340,@#6
050466  005737  177766                  TST     @#177766        ;CHECK TO SEE IF THE POWER FAILED,WILL TRAP
                                                                ;TO 1$ IF UNIBUS MAP BOARD IS NOT PRESENT
                                                                ;SINCE THE CPUERR(177766)IS ON THE UNIBUS
                                                                ;MAP BOARD
050472  001404                          BEQ     1$              ;NO,GO REPORT THE ERROR
050474  012737  000177  001002          MOV     #177,@#$FATAL   ;YES,FLAG THAT THE POWER FAILED
050502  000403                          BR      2$              ; AND GO SET FATAL ERROR FLAG
050504  012737  000002  001002  1$:     MOV     #2,@#$FATAL                     ;SET UP FATAL ERROR NUMBER
050512  012767  000001  130260  2$:     MOV     #1,$MSGTY                       ;SET FATAL ERROR FLAG
050520  000777                  MMUHLT: BR      .               ;STAY IN LOOP
```

```
                000244                          FPVECT=244
                                                        .SBTTL  FPP REGISTER DEFINITIONS
                000000                          AC0     =%0
                000001                          AC1     =%1
                000002                          AC2     =%2
                000003                          AC3     =%3
                000004                          AC4     =%4
                000005                          AC5     =%5
                000006                          AC6     =%6
                000007                          AC7     =%7

        050522  012706  001000                  FPSTRT: MOV     #STBOT,SP               ;SET UP STACK POINTER
        050526  032777  000002  150570                  BIT     #2,@SWR
        050534  001402                                  BEQ     1$
        050536  000167  050016                          JMP     SLU1ST
        050542  012737  120424  000030          1$:     MOV     #ERROR4,@#30            ;SETUP FOR CORRECT ERROR CALL
        050550  012737  000003  001004                  MOV     #3,@#$TESTN             ;PUT TEST NUMBER IN MAILBOX

                                                ;*************************************************************************
                                                ;TEST 420       LDFPS, STFPS AND DATA PATHS TEST
                                                ;*************************************************************************
        050556                                  TS420:
        050556  012700  177777                          MOV     #-1,R0                          ;INITIALIZE THE COUNT PATTERN.
        050562  012737  050634  000244                  MOV     #AERR1,@#FPVECT                 ;SET UP FOR UNABLE TO DECODE
        050570  012737  050634  000010                  MOV     #AERR1,@#10                     ;FPP INSTRUCTION TRAP TO 244 OR 10.
        050576  012737  050634  000004                  MOV     #AERR1,@#ERRVECT                ;IF EITHER INSTRUCTION
                                                                                                ;FAILS TO GO THROUGH THE
                                                                                                ;CORRECT SRC OR DST MODE AN
                                                                                                ;ODD ADDRESS TRAP WILL OCCUR.
        050604                                  A1:
        050604  010004                          A11:    MOV     R0,R4
        050606  042704  030020                          BIC     #30020,R4
        050612  170104                                  LDFPS   R4                              ;TEST INSTRUCTION.

        050614  012701  177777                          MOV     #-1,R1
        050620  170201                          A12:    STFPS   R1                              ;TEST INSTRUCTION.
        050622  010004                                  MOV     R0,R4                           ;MASK OFF UNSETTABLE BITS.
        050624  042704  030020                          BIC     #30020,R4
        050630  020401                                  CMP     R4,R1                           ;COMPARE DATA EXPECTED WITH
                                                                                                ;THE DATA READ.
        050632  001401                                  BEQ     A2
        050634                                  AERR1:
        050634  104000                                  EMT                     ;

        050636  012700  000001                  A2:     MOV     #1,R0                           ;NEXT PATTERN WILL BE ALL ZERO
        050642  077020                                  SOB     R0,A1                           ;DECREMENT COUNT PATTERN
        050644                                  ADONE:
```

```
050644  004767  047624          JSR     PC,.RSET        ;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?).

                                ;*******************************************************************
                                ;TEST 421       CFCC TEST
                                ;*******************************************************************
050650                          TS421:
050650  012700  000017                  MOV     #17,R0                  ;R0 CONTAINS TO TEST PATTERN.

050654                          B1:
050654  170100                          LDFPS   R0                      ;LOAD THE TEST PATTERN

050656                          B2:
050656  170000                          CFCC                            ;COPY CONDITION CODES.

050660  013703  177776                  MOV     @#PSW,R3                ;SEE IF PATTERN TRANSFERED.
050664  042703  177760                  BIC     #177760,R3
050670  020003                          CMP     R0,R3
050672  001401                          BEQ     B3
050674  104000                          EMT                     ;
050676  077012                  B3:     SOB     R0,B1
050700                          BDONE:
050700  004767  047570                  JSR     PC,.RSET        ;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?).

                                ;*******************************************************************
                                ;TEST 422       SETF, SETD, SETI AND SETL TEST
                                ;*******************************************************************
050704                          TS422:
050704  005000                          CLR     R0

050706  170100                          LDFPS   R0                      ;CLEAR THE FPS.
050710  170001                  C15:    SETF                            ;TEST INSTRUCTION.

050712  170201                          STFPS   R1                      ;GET RESULT.
050714  005002                          CLR     R2
050716  020201                          CMP     R2,R1                   ;DID AN ERROR OCCUR?
050720  001401                          BEQ     C2
050722  104000                          EMT                     ;
050724  012700  147757          C2:     MOV     #147757,R0

050730  170100                          LDFPS   R0                      ;PUT 147757 IS FPS
050732  170001                  C25:    SETF                            ;CLEAR FD BIT.

050734  170201                          STFPS   R1                      ;GET RESULT
050736  012702  147557                  MOV     #147557,R2
050742  020102                          CMP     R1,R2                   ;RESULT CORRECT.
050744  001401                          BEQ     C3
050746  104000                          EMT                     ;
```

```
050750  012700  147757   C3:    MOV     #147757,R0
050754  170100                  LDFPS   R0                      ;LOAD 147757 INTO FPS.
050756  170011           C35:   SETD                            ;SETD FD BIT.
050760  170201                  STFPS   R1
050762  012702  147757          MOV     #147757,R2
050766  020102                  CMP     R1,R2                   ;RESULT CORRECT?
050770  001401                  BEQ     C4
050772  104000                  EMT                     ;
050774  005000           C4:    CLR     R0
050776  170100                  LDFPS   R0                      ;CLEAR FPS.
051000  170011           C45:   SETD                            ;SET FD BIT.
051002  170201                  STFPS   R1                      ;GET RESULT.
051004  012702  000200          MOV     #200,R2
051010  020102                  CMP     R1,R2                   ;RESULT CORRECT?
051012  001401                  BEQ     C5
051014  104000                  EMT                     ;
051016  005000           C5:    CLR     R0
051020  170100                  LDFPS   R0                      ;CLEAR FPS
051022  170002           C55:   SETI                            ;CLEAR FL BIT.
051024  170201                  STFPS   R1                      ;GET RESULT.
051026  005002                  CLR     R2
051030  020201                  CMP     R2,R1                   ;RESULT CORRECT?
051032  001401                  BEQ     C6
051034  104000                  EMT                     ;
051036  012700  147757   C6:    MOV     #147757,R0
051042  170100                  LDFPS   R0                      ;PUT 147757 INTO FPS
051044  170002           C65:   SETI                            ;CLEAR FL BIT.
051046  170201                  STFPS   R1                      ;GET THE RESULT.
051050  012702  147657          MOV     #147657,R2
051054  020102                  CMP     R1,R2                   ;RESULT CORRECT?
051056  001401                  BEQ     C7
051060  104000                  EMT                     ;
051062  012700  147757   C7:    MOV     #147757,R0
051066  170100                  LDFPS   R0                      ;SET FPS TO 147757.
051070  170012           C75:   SETL                            ;SET FL BIT.
051072  170201                  STFPS   R1                      ;GET THE RESULT.
051074  012702  147757          MOV     #147757,R2
051100  020102                  CMP     R1,R2                   ;RESULT CORRECT?
051102  001401                  BEQ     C8
051104  104000                  EMT                     ;
051106  005000           C8:    CLR     R0
051110  170100                  LDFPS   R0                      ;CLEAR FPS.
051112  170012           C85:   SETL                            ;SET FL BIT.
051114  170201                  STFPS   R1
051116  012702  000100          MOV     #100,R2
051122  020102                  CMP     R1,R2                   ;RESULT CORRECT.
051124  001401                  BEQ     CDONE
051126  104000                  EMT                     ;
```

```
051130                            CDONE:
051130  004767  047340                    JSR     PC,.RSET        ;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?).


                                  ;*******************************************************************
                                  ;TEST 423        ILLEGAL FPP OP CODES AND STST TEST
                                  ;*******************************************************************
051134                            TS423:
051134  012705  170003                    MOV     #170003,R5              ;INITIAL OP CODE.
051140  012737  051200  000004            MOV     #DERR2,@#ERRVECT
051146  012737  051232  000244            MOV     #DERR1,@#FPVECT

051154  005000                    D1:     CLR     R0
051156  170100                            LDFPS   R0                      ;CLEAR FPS.
051160  005002                            CLR     R2
051162  010537  051166                    MOV     R5,@#D2                 ;SET UP THE ILLEGAL INSTRUCTION.
051166  000000                    D2:     .WORD   0
051170  170000                    D3:     CFCC
051172  005202                            INC     R2
051174  005202                    D4:     INC     R2

051176  170201                            STFPS   R1                      ;REPORT FAILURE. DID NOT TRAP.
051200                            DERR2:
051200  104000                            EMT                     ;
051202  022705  170010            D5:     CMP     #170010,R5              ;COMPUTE NEXT OP CODE
051206  001003                            BNE     D6
051210  012705  170013                    MOV     #170013,R5
051214  000757                            BR      D1

051216  022705  170077            D6:     CMP     #170077,R5
051222  001001                            BNE     D7
051224  000424                            BR      DDONE
051226  005205                    D7:     INC     R5
051230  000751                            BR      D1

051232  022716  051170            DERR1:  CMP     #D3,(SP)                ;DID TRAP OCCUR ON TEST INSTRUCTION?
051236  001401                            BEQ     1$
051240  104000                            EMT                     ;
051242  022626                    1$:     CMP     (SP)+,(SP)+
051244  170201                            STFPS   R1                      ;GET THE FPS AND SEE IF IT IS
051246  022701  100000                    CMP     #100000,R1              ;SET CORRECTLY.
051252  001401                            BEQ     3$
051254  104000                            EMT                     ;
051256  012704  000001            3$:     MOV     #1,R4
051262  170304                    D8:     STST    R4                      ;GET THE FEC CODE. NOTE THAT
                                                                          ;IF THE DESTINATION MODE IS
                                                                          ;IMPROPERLY DECODED AN ODD
                                                                          ;ADDRESS TRAP TO 4 SHOULD OCCUR.
051264  022704  000002                    CMP     #2,R4                   ;WAS FEC CORRECT?
051270  001001                            BNE     D9
051272  000743                            BR      D5

```

```
051274                          D9:                                     ;REPORT STST FAILURE
051274  104000                          EMT                     ;
051276                          DDONE:
051276  004767  047172                  JSR     PC,.RSET        ;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?).

                                ;*******************************************************************************
                                ;TEST 424       FID, INTERRUPT DISABLE, BIT TEST
                                ;*******************************************************************************
051302                          TS424:
051302  012737  051342  000244          MOV     #EERRO,@#FPVECT         ;SETUP FOR THE INTERRUPT.

051310  012700  040000          E1:     MOV     #40000,R0
051314  170100                          LDFPS   R0                      ;SET FID.
051316  170020                  E3:     .WORD   170020                  ;ILLEGAL FPP INSTRUCTION.
051320  170000                  E4:     CFCC

051322  170201                          STFPS   R1                      ;SEE IF ERROR WAS DETECTED.
051324  022701  140000                  CMP     #140000,R1
051330  001004                          BNE     EERRO

051332  170304                          STST    R4                      ;SEE IF FEC=2
051334  022704  000002                  CMP     #2,R4
051340  001401                          BEQ     EDONE
051342                          EERRO:
051342  104000                          EMT                     ;
051344                          EDONE:
051344  004767  047124                  JSR     PC,.RSET        ;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?).


                                ;*******************************************************************************
                                ;TEST 425       LDD AND STD, WITH SRC AND DST MODE 1, TEST
                                ;*******************************************************************************
051350                          TS425:

051350  005000                          CLR     R0
051352  170100                          LDFPS   R0
051354  170011                          SETD
051356  012701  051620                  MOV     #FDATI0,R1              ;SET UP THE LOAD DATA.
051362  012702  051664                  MOV     #FXDAT0,R2
051366  012703  000010                  MOV     #10,R3

051372  012221                  F2:     MOV     (R2)+,(R1)+
051374  077302                          SOB     R3,F2

051376  012700  051630                  MOV     #FDATI4,R0              ;SETUP R0 FOR THE LDD (R0),AC0.
051402  012737  051616  000004          MOV     #FERR20,@#ERRVECT       ;IF THE SRC FLOWS FAIL THEN
                                                                        ;AN ODD ADDRESS MAY OCCUR.
051410  005003                          CLR     R3
```

```
051412  172410                  F3:     LDD     (R0),AC0
051414  005203                  F4:     INC     R3
051416  005203                          INC     R3
051420  020027  051630                  CMP     R0,#FDATI4              ;WAS R0 AFFECTED?
051424  001401                          BEQ     F5
051426  104000                          EMT             ;
051430  020327  000002          F5:     CMP     R3,#2                   ;SEE IF THE PC WAS ADVERSELY
051434  001401                          BEQ     1$
051436  104000                          EMT             ;
051440  012701  051620          1$:     MOV     #FDATI0,R1              ;MAKE SURE THE SOURCE DATA WAS
051444  012702  051664                  MOV     #FXDAT0,R2              ;NOT AFFECTED.
051450  012703  000010                  MOV     #10,R3
051454  022122                  2$:     CMP     (R1)+,(R2)+
051456  001401                          BEQ     3$
051460  104000                          EMT             ;
051462  077304                  3$:     SOB     R3,2$
051464  170201                          STFPS   R1                      ;MAKE SURE THE FPS IS CORRECT.
051466  022701  000200                  CMP     #200,R1
051472  001401                          BEQ     F6
051474  104000                          EMT             ;
051476  012703  177777          F6:     MOV     #-1,R3
051502  012704  000010                  MOV     #10,R4
051506  012705  051642                  MOV     #FDAT00,R5              ;SET UP THE OUTPUT DATA BUFFER.
051512  010325                  F7:     MOV     R3,(R5)+
051514  077402                          SOB     R4,F7
051516  012700  051652                  MOV     #FDAT04,R0              ;SET UP R0 FOR DST MODE 1 REG 0.
051522  012737  051616  000004          MOV     #FERR20,@#ERRVECT       ;IF THE DST FLOWS FAIL AN ODD
                                                                        ;ADDRESS COULD OCCUR.
051530  005003                          CLR     R3
051532  174010                  F10:    STD     AC0,(R0)                ;TEST INSTRUCTION.
051534  005203                  F11:    INC     R3
051536  005203                          INC     R3
051540  020027  051652                  CMP     R0,#FDAT04              ;WAS R0 MODIFIED?
051544  001401                          BEQ     F12
051546  104000                          EMT             ;
051550  020327  000002          F12:    CMP     R3,#2                   ;WAS THE PC AFFECTED CORRECTLY?
051554  001401                          BEQ     F135
051556  104000                          EMT             ;
051560  012701  051642          F135:   MOV     #FDAT00,R1
051564  012702  051664                  MOV     #FXDAT0,R2
051570  012703  000010                  MOV     #10,R3          ;SETUP LOOP COUNT
051574  022122                  F13:    CMP     (R1)+,(R2)+     ;WAS DATA OUTPUT CORRECTLY
051576  001401                          BEQ     F14
051600  104000                          EMT             ;
051602  077304                  F14:    SOB     R3,F13          ;SUBTRACT 1 FROM LOOP COUNT AND LOOP IF NOT ZERO
051604  005001                  F22:    CLR     R1
051606  170201                          STFPS   R1                      ;MAKE SURE FPS IS CORRECT.
051610  022701  000200                  CMP     #200,R1
051614  001433                          BEQ     FDONE
051616                          FERR20:
```

```
051616  104000                  EMT                             ;

051620  177777          FDATI0: -1
051622  177777          FDATI1: -1
051624  177777          FDATI2: -1
051626  177777          FDATI3: -1
051630  177777          FDATI4: -1
051632  177777          FDATI5: -1
051634  177777          FDATI6: -1
051636  177777          FDATI7: -1
051640  177777                  -1
051642  177777          FDATO0: -1
051644  177777          FDATO1: -1
051646  177777          FDATO2: -1
051650  177777          FDATO3: -1
051652  177777          FDATO4: -1
051654  177777          FDATO5: -1
051656  177777          FDATO6: -1
051660  177777          FDATO7: -1
051662  177777                  -1
051664  177777          FXDAT0: -1
051666  177777          FXDAT1: -1
051670  177777          FXDAT2: -1
051672  177777          FXDAT3: -1
051674  052525          FXDAT4: 052525
051676  031463          FXDAT5: 031463
051700  007417          FXDAT6: 007417
051702  000477          FXDAT7: 000477


051704                  FDONE:
051704  004767  046564          JSR     PC,.RSET        ;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?).



                        ;*******************************************************************
                        ;TEST 426       FSRC MODE 0 TEST
                        ;*******************************************************************
051710                  TS426:
051710                  I1:
051710  170011                  SETD                            ;SET FD.
051712  012700  052176          MOV     #IDATI0,R0
051716  012701  052146          MOV     #IPATI0,R1
051722  012702  000004          MOV     #4,R2
051726  012120          I2:     MOV     (R1)+,(R0)+             ;SET UP THE INPUT DATA BUFFER.
051730  077202                  SOB     R2,I2

051732  012700  052176          MOV     #IDATI0,R0              ;LOAD AC1
051736  172510                  LDD     (R0),AC1

051740  012700  052156          MOV     #IPAT20,R0              ;LOAD AC0
051744  172410                  LDD     (R0),AC0

```

```
051746  012701  000001                  MOV     #1,R1                   ;IN CASE THE FSRC FLOWS FAIL
051752  172401                  I3:     LDD     AC1,AC0                 ;TEST INSTRUCTION.
051754  000240                  I4:     NOP
051756  000240                  I5:     NOP

051760  012700  052166                  MOV     #IDAT00,R0
051764  174010                          STD     AC0,(R0)                ;GET AC0, THE RESULTS.

051766  012700  052166                  MOV     #IDAT00,R0              ;SEE IF DATA IS CORRECT.
051772  012701  052176                  MOV     #IDAT10,R1
051776  012702  000004                  MOV     #4,R2
052002  022021                  I6:     CMP     (R0)+,(R1)+
052004  001401                          BEQ     I105
052006  104000                          EMT                     ;
052010  077204                  I105:   SOB     R2,I6

                                ;NOW TEST THE LOAD INSTRUCTION WITH FSRC MODE ZERO AND FD CLEAR.

052012  012700  052146          I12:    MOV     #IPAT10,R0
052016  012701  052176                  MOV     #IDAT10,R1
052022  012702  000004                  MOV     #4,R2
052026  012021                  I13:    MOV     (R0)+,(R1)+
052030  077202                          SOB     R2,I13

052032  012700  052176                  MOV     #IDAT10,R0              ;SET UP AC1
052036  172510                          LDD     (R0),AC1

052040  012700  052156                  MOV     #IPAT20,R0              ;SET UP AC0
052044  172410                          LDD     (R0),AC0

052046  012701  000001                  MOV     #1,R1
052052  170001                          SETF                            ;CLEAR FD.

052054  172401                  I14:    LDF     AC1,AC0                 ;TEST INSTRUCTION.
052056  000240                  I15:    NOP
052060  000240                  I16:    NOP

052062  170200                          STFPS   R0                      ;SEE IF FPS IS STILL CLEAR.
052064  022700  000004                  CMP     #4,R0
052070  001401                          BEQ     I17
052072  104000                          EMT                     ;
052074                          I17:                                    ;RESET TO DOUBLE MODE.
052074  170011                          SETD

052076  012700  052166                  MOV     #IDAT00,R0
052102  174010                          STD     AC0,(R0)                ;GET AC0

052104  012737  177777  052202          MOV     #-1,@#IDAT12
052112  012737  177777  052204          MOV     #-1,@#IDAT13
052120  012700  052166                  MOV     #IDAT00,R0
052124  012701  052176                  MOV     #IDAT10,R1
052130  012702  000004                  MOV     #4,R2
052134  022021                  I20:    CMP     (R0)+,(R1)+             ;SEE IF AC0 WAS CORRECT.
052136  001401                          BEQ     I23
052140  104000                          EMT                     ;
052142  077204                  I23:    SOB     R2,I20
```

```
052144  000420                  BR      IDONE                   ;NO ERRORS.

052146  000000          IPAT10: 0
052150  170360          IPAT11: 170360
052152  016161          IPAT12: 016161
052154  052525          IPAT13: 052525

052156  177777          IPAT20: -1
052160  177777          IPAT21: -1
052162  177777          IPAT22: -1
052164  177777          IPAT23: -1

052166  000000          IDAT00: 0
052170  000000          IDAT01: 0
052172  000000          IDAT02: 0
052174  000000          IDAT03: 0

052176  000000          IDAT10: 0
052200  000000          IDAT11: 0
052202  000000          IDAT12: 0
052204  000000          IDAT13: 0

052206                  IDONE:
052206  004767  046262          JSR     PC,.RSET        ;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?).



                        ;*************************************   ****************************************
                        ;TEST 427       FDST MODE 0 TEST
                        ;*************************************   ****************************************
052212                  TS427:
052212  170011                  SETD                            ;SET FD
052214  012700  052452          MOV     #TPAT10,R0
052220  012701  052502          MOV     #TDAT10,R1
052224  012702  000004          MOV     #4,R2
052230  012021          T2:     MOV     (R0)+,(R1)+             ;SET UP THE INPUT DATA BUFFER.
052232  077202                  SOB     R2,T2

052234  012700  052502          MOV     #TDAT10,R0              ;LOAD AC0
052240  172410                  LDD     (R0),AC0

052242  012700  052462          MOV     #TPAT20,R0              ;LOAD AC1
052246  172510                  LDD     (R0),AC1

052250  012701  000001          MOV     #1,R1                   ;IF THE (BUT FDST) FORK FAILS
052254  174001          T3:     STD     AC0,AC1
052256  000240          T4:     NOP
052260  000240          T5:     NOP

052262  012700  052472          MOV     #TDAT00,R0
052266  174110                  STD     AC1,(R0)                ;GET THE DATA.

052270  012703  052472          MOV     #TDAT00,R3              ;SEE IF THE DATA IS CORRECT.
```

```
052274  012704  052502                  MOV     #TDATI0,R4
052300  012705  000004                  MOV     #4,R5
052304  022324                  T6:     CMP     (R3)+,(R4)+
052306  001401                          BEQ     T105
052310  104000                          EMT                     ;
052312  077504                  T105:   SOB     R5,T6

                                .NOW TEST THE STF AC0,AC1 INSTRUCTION.

052314  012700  052452          T12:    MOV     #TPAT10,R0              ;SET UP THE INPUT DATA BUFFER.
052320  012701  052502                  MOV     #TDATI0,R1
052324  012702  000004                  MOV     #4,R2
052330  012021                  T13:    MOV     (R0)+,(R1)+
052332  077202                          SOB     R2,T13

052334  012700  052502                  MOV     #TDATI0,R0              ;SET UP AC0
052340  172410                          LDD     (R0),AC0

052342  012700  052462                  MOV     #TPAT20,R0              ;SET UP AC1
052346  172510                          LDD     (R0),AC1

052350  012701  000001                  MOV     #1,R1
052354  170001                          SETF                            ;CLEAR FD
052356  174001                  T14:    STF     AC0,AC1
052360  000240                  T15:    NOP
052362  000240                  T16:    NOP

052364  005000                          CLR     R0
052366  170200                          STFPS   R0                      ;SEE IF FPS IS CLEAR.
052370  022700  000010                  CMP     #10,R0
052374  001401                          BEQ     T17
052376  104000                          EMT                     ;
052400                          T17:
052400  170011                          SETD                            ;SET FD.

052402  012700  052472                  MOV     #TDAT00,R0
052406  174110                          STD     AC1,(R0)                ;PICK UP AC1.

052410  012737  177777  052506          MOV     #-1,@#TDAT12
052416  012737  177777  052510          MOV     #-1,@#TDAT13
052424  012703  052472                  MOV     #TDAT00,R3
052430  012704  052502                  MOV     #TDATI0,R4
052434  012705  000004                  MOV     #4,R5
052440  022324                  T20:    CMP     (R3)+,(R4)+             ;WAS THE DATA TRANSFERRED CORRECTLY?
052442  001401                          BEQ     T23
052444  104000                          EMT                     ;
052446  077504                  T23:    SOB     R5,T20
052450  000420                          BR      TDONE


052452  000000                  TPAT10: 0
052454  170360                  TPAT11: 170360
052456  016161                  TPAT12: 016161
052460  052525                  TPAT13: 052525

052462  177777                  TPAT20: -1
```

```
052464  177777                  TPAT21: -1
052466  177777                  TPAT22: -1
052470  177777                  TPAT23: -1

052472  000000                  TDAT00: 0
052474  000000                  TDAT01: 0
052476  000000                  TDAT02: 0
052500  000000                  TDAT03: 0

052502  000000                  TDATI0: 0
052504  000000                  TDATI1: 0
052506  000000                  TDATI2: 0
052510  000000                  TDATI3: 0

052512                          TDONE:
052512  004767  045756                  JSR     PC,.RSET        ;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?).



                                ;*******************************************************************
                                ;TEST 430       ACCUMULATORS DATA PATTERNS TEST
                                ;*******************************************************************
052516                          TS430:
052516  170011                          SETD                            ;SET FD.
                                ;TEST ACCUMULATOR 0 WITH FLOATING ONE
052520  012700  054252                  MOV     #GPAT00,R0
052524  012701  054312                  MOV     #GDAT00,R1
052530  004737  054124                  JSR     PC,@#GSETUP             ;LOAD TEST PATTERN.
052534  012703  000102                  MOV     #102,R3
052540                          G1:
052540  172410                          LDD     (R0),AC0
052542  174000                          STD     AC0,AC0
052544  172400                          LDD     AC0,AC0         ;STORE THE TEST PATTERN.
052546  174011                          STD     AC0,(R1)
052550  004737  054222                  JSR     PC,@#GCMP               ;COMPARE THE DATA READ WITH
                                                                        ;THAT WHICH WAS WRITTEN.
052554  005737  054246                  TST     @#GFLAG1
052560  001004                          BNE     G2
052562  005137  054246                  COM     @#GFLAG1
052566  000261                          SEC
052570  000401                          BR      G3
052572  000241                  G2:     CLC
052574  006160  000006          G3:     ROL     6(R0)                   ;GENERATE THE NEXT TEST PATTERN.
052600  006160  000004                  ROL     4(R0)
052604  006160  000002                  ROL     2(R0)
052610  006110                          ROL     (R0)
052612  004737  054202                  JSR     PC,@#GRESET             ;RESET DEFAULT PATTERN IN OUTPUT
                                                                        ;BUFFER.
052616  077330                          SOB     R3,G1


                                ;TEST ACCUMULATOR 0 WITH FLOATING ZERO
052620  012700  054262                  MOV     #GPAT10,R0
```

```
052624 012701 054312          MOV     #GDAT00,R1
052630 004737 054124          JSR     PC,@#GSETUP              ;LOAD TEST PATTERN.
052634 012703 000102          MOV     #102,R3
052640                 G4:
052640 172410                 LDD     (R0),AC0
052642 174000                 STD     AC0,AC0
052644 172400                 LDD     AC0,AC0         ;STORE THE TEST PATTERN.
052646 174011                 STD     AC0,(R1)
052650 004737 054222          JSR     PC,@#GCMP                ;COMPARE THE DATA READ WITH
                                                               ;THAT WHICH WAS WRITTEN.
052654 005737 054246          TST     @#GFLAG1
052660 001004                 BNE     G5
052662 005137 054246          COM     @#GFLAG1
052666 000241                 CLC
052670 000401                 BR      G6
052672 000261          G5:    SEC
052674 006160 000006   G6:    ROL     6(R0)                    ;GENERATE THE NEXT TEST PATTERN.
052700 006160 000004          ROL     4(R0)
052704 006160 000002          ROL     2(R0)
052710 006110                 ROL     (R0)
052712 004737 054202          JSR     PC,@#GRESET              ;RESET DEFAULT PATTERN IN OUTPUT
                                                               ;BUFFER.
052716 077330                 SOB     R3,G4

                       ;TEST ACCUMULATOR 1 WITH FLOATING ONE
052720 012700 054252          MOV     #GPAT00,R0
052724 012701 054312          MOV     #GDAT00,R1
052730 004737 054124          JSR     PC,@#GSETUP              ;LOAD TEST PATTERN.
052734 012703 000102          MOV     #102,R3
052740                 G7:
052740 172410                 LDD     (R0),AC0
052742 174001                 STD     AC0,AC1
052744 172401                 LDD     AC1,AC0         ;STORE THE TEST PATTERN.
052746 174011                 STD     AC0,(R1)
052750 004737 054222          JSR     PC,@#GCMP                ;COMPARE THE DATA READ WITH
                                                               ;THAT WHICH WAS WRITTEN.
052754 005737 054246          TST     @#GFLAG1
052760 001004                 BNE     G10
052762 005137 054246          COM     @#GFLAG1
052766 000261                 SEC
052770 000401                 BR      G11
052772 000241          G10:   CLC
052774 006160 000006   G11:   ROL     6(R0)                    ;GENERATE THE NEXT TEST PATTERN.
053000 006160 000004          ROL     4(R0)
053004 006160 000002          ROL     2(R0)
053010 006110                 ROL     (R0)
053012 004737 054202          JSR     PC,@#GRESET              ;RESET DEFAULT PATTERN IN OUTPUT
                                                               ;BUFFER.
053016 077330                 SOB     R3,G7

                       ;TEST ACCUMULATOR 1 WITH FLOATING ZERO
053020 012700 054262          MOV     #GPAT10,R0
053024 012701 054312          MOV     #GDAT00,R1
053030 004737 054124          JSR     PC,@#GSETUP              ;LOAD TEST PATTERN.
053034 012703 000102          MOV     #102,R3
053040                 G12:
```

```
053040  172410                   LDD     (R0),AC0
053042  174001                   STD     AC0,AC1
053044  172401                   LDD     AC1,AC0          ;STORE THE TEST PATTERN.
053046  174011                   STD     AC0,(R1)
053050  004737  054222           JSR     PC,@#GCMP                 ;COMPARE THE DATA READ WITH
                                                                   ;THAT WHICH WAS WRITTEN.
053054  005737  054246           TST     @#GFLAG1
053060  001004                   BNE     G13
053062  005137  054246           COM     @#GFLAG1
053066  000241                   CLC
053070  000401                   BR      G14
053072  000261           G13:    SEC
053074  006160  000006   G14:    ROL     6(R0)                     ;GENERATE THE NEXT TEST PATTERN.
053100  006160  000004           ROL     4(R0)
053104  006160  000002           ROL     2(R0)
053110  006110                   ROL     (R0)
053112  004737  054202           JSR     PC,@#GRESET               ;RESET DEFAULT PATTERN IN OUTPUT
                                                                   ;BUFFER.
053116  077330                   SOB     R3,G12

                         ;TEST ACCUMULATOR 2 WITH FLOATING ONE
053120  012700  054252           MOV     #GPAT00,R0
053124  012701  054312           MOV     #GDAT00,R1
053130  004737  054124           JSR     PC,@#GSETUP               ;LOAD TEST PATTERN.
053134  012703  000102           MOV     #102,R3
053140                   G15:
053140  172410                   LDD     (R0),AC0
053142  174002                   STD     AC0,AC2
053144  172402                   LDD     AC2,AC0          ;STORE THE TEST PATTERN.
053146  174011                   STD     AC0,(R1)
053150  004737  054222           JSR     PC,@#GCMP                 ;COMPARE THE DATA READ WITH
                                                                   ;THAT WHICH WAS WRITTEN.
053154  005737  054246           TST     @#GFLAG1
053160  001004                   BNE     G16
053162  005137  054246           COM     @#GFLAG1
053166  000261                   SEC
053170  000401                   BR      G17
053172  000241           G16:    CLC
053174  006160  000006   G17:    ROL     6(R0)                     ;GENERATE THE NEXT TEST PATTERN.
053200  006160  000004           ROL     4(R0)
053204  006160  000002           ROL     2(R0)
053210  006110                   ROL     (R0)
053212  004737  054202           JSR     PC,@#GRESET               ;RESET DEFAULT PATTERN IN OUTPUT
                                                                   ;BUFFER.
053216  077330                   SOB     R3,G15

                         ;TEST ACCUMULATOR 2 WITH FLOATING ZERO
053220  012700  054262           MOV     #GPAT10,R0
053224  012701  054312           MOV     #GDAT00,R1
053230  004737  054124           JSR     PC,@#GSETUP               ;LOAD TEST PATTERN.
053234  012703  000102           MOV     #102,R3
053240                   G20:
053240  172410                   LDD     (R0),AC0
053242  174002                   STD     AC0,AC2
053244  172402                   LDD     AC2,AC0          ;STORE THE TEST PATTERN.
053246  174011                   STD     AC0,(R1)
```

```
053250 004737 054222           JSR     PC,@#GCMP               ;COMPARE THE DATA READ WITH
                                                               ;THAT WHICH WAS WRITTEN.
053254 005737 054246           TST     @#GFLAG1
053260 001004                  BNE     G21
053262 005137 054246           COM     @#GFLAG1
053266 000241                  CLC
053270 000401                  BR      G22
053272 000261           G21:   SEC
053274 006160 000006    G22:   ROL     6(R0)                   ;GENERATE THE NEXT TEST PATTERN.
053300 006160 000004           ROL     4(R0)
053304 006160 000002           ROL     2(R0)
053310 006110                  ROL     (R0)
053312 004737 054202           JSR     PC,@#GRESET             ;RESET DEFAULT PATTERN IN OUTPUT
                                                               ;BUFFER.
053316 077330                  SOB     R3,G20

                        ;TEST ACCUMULATOR 3 WITH FLOATING ONE
053320 012700 054252           MOV     #GPAT00,R0
053324 012701 054312           MOV     #GDAT00,R1
053330 004737 054124           JSR     PC,@#GSETUP             ;LOAD TEST PATTERN.
053334 012703 000102           MOV     #102,R3
053340                  G23:
053340 172410                  LDD     (R0),AC0
053342 174003                  STD     AC0,AC3
053344 172403                  LDD     AC3,AC0         ;STORE THE TEST PATTERN.
053346 174011                  STD     AC0,(R1)
053350 004737 054222           JSR     PC,@#GCMP               ;COMPARE THE DATA READ WITH
                                                               ;THAT WHICH WAS WRITTEN.
053354 005737 054246           TST     @#GFLAG1
053360 001004                  BNE     G24
053362 005137 054246           COM     @#GFLAG1
053366 000261                  SEC
053370 000401                  BR      G25
053372 000241           G24:   CLC
053374 006160 000006    G25:   ROL     6(R0)                   ;GENERATE THE NEXT TEST PATTERN.
053400 006160 000004           ROL     4(R0)
053404 006160 000002           ROL     2(R0)
053410 006110                  ROL     (R0)
053412 004737 054202           JSR     PC,@#GRESET             ;RESET DEFAULT PATTERN IN OUTPUT
                                                               ;BUFFER.
053416 077330                  SOB     R3,G23

                        ;TEST ACCUMULATOR 3 WITH FLOATING ZERO
053420 012700 054262           MOV     #GPAT10,R0
053424 012701 054312           MOV     #GDAT00,R1
053430 004737 054124           JSR     PC,@#GSETUP             ;LOAD TEST PATTERN.
053434 012703 000102           MOV     #102,R3
053440                  G26:
053440 172410                  LDD     (R0),AC0
053442 174003                  STD     AC0,AC3
053444 172403                  LDD     AC3,AC0         ;STORE THE TEST PATTERN.
053446 174011                  STD     AC0,(R1)
053450 004737 054222           JSR     PC,@#GCMP               ;COMPARE THE DATA READ WITH
                                                               ;THAT WHICH WAS WRITTEN.
053454 005737 054246           TST     @#GFLAG1
053460 001004                  BNE     G27
```

```
053462  005137  054246          COM     @#GFLAG1
053466  000241                  CLC
053470  000401                  BR      G30
053472  000261          G27:    SEC
053474  006160  000006  G30:    ROL     6(R0)                   ;GENERATE THE NEXT TEST PATTERN.
053500  006160  000004          ROL     4(R0)
053504  006160  000002          ROL     2(R0)
053510  006110                  ROL     (R0)
053512  004737  054202          JSR     PC,@#GRESET             ;RESET DEFAULT PATTERN IN OUTPUT
                                                                ;BUFFER.
053516  077330                  SOB     R3,G26


                        ;TEST ACCUMULATOR 4 WITH FLOATING ONE
053520  012700  054252          MOV     #GPAT00,R0
053524  012701  054312          MOV     #GDAT00,R1
053530  004737  054124          JSR     PC,@#GSETUP             ;LOAD TEST PATTERN.
053534  012703  000102          MOV     #102,R3
053540                  G31:
053540  172410                  LDD     (R0),AC0
053542  174004                  STD     AC0,AC4
053544  172404                  LDD     AC4,AC0         ;STORE THE TEST PATTERN.
053546  174011                  STD     AC0,(R1)
053550  004737  054222          JSR     PC,@#GCMP               ;COMPARE THE DATA READ WITH
                                                                ;THAT WHICH WAS WRITTEN.
053554  005737  054246          TST     @#GFLAG1
053560  001004                  BNE     G32
053562  005137  054246          COM     @#GFLAG1
053566  000261                  SEC
053570  000401                  BR      G33
053572  000241          G32:    CLC
053574  006160  000006  G33:    ROL     6(R0)                   ;GENERATE THE NEXT TEST PATTERN.
053600  006160  000004          ROL     4(R0)
053604  006160  000002          ROL     2(R0)
053610  006110                  ROL     (R0)
053612  004737  054202          JSR     PC,@#GRESET             ;RESET DEFAULT PATTERN IN OUTPUT
                                                                ;BUFFER.
053616  077330                  SOB     R3,G31


                        ;TEST ACCUMULATOR 4 WITH FLOATING ZERO
053620  012700  054262          MOV     #GPAT10,R0
053624  012701  054312          MOV     #GDAT00,R1
053630  004737  054124          JSR     PC,@#GSETUP             ;LOAD TEST PATTERN.
053634  012703  000102          MOV     #102,R3
053640                  G34:
053640  172410                  LDD     (R0),AC0
053642  174004                  STD     AC0,AC4
053644  172404                  LDD     AC4,AC0         ;STORE THE TEST PATTERN.
053646  174011                  STD     AC0,(R1)
053650  004737  054222          JSR     PC,@#GCMP               ;COMPARE THE DATA READ WITH
                                                                ;THAT WHICH WAS WRITTEN.
053654  005737  054246          TST     @#GFLAG1
053660  001004                  BNE     G35
053662  005137  054246          COM     @#GFLAG1
053666  000241                  CLC
053670  000401                  BR      G36
053672  000261          G35:    SEC
```

```
053674  006160  000006   G36:  ROL   6(R0)              ;GENERATE THE NEXT TEST PATTERN.
053700  006160  000004         ROL   4(R0)
053704  006160  000002         ROL   2(R0)
053710  006110                 ROL   (R0)
053712  004737  054202         JSR   PC,@#GRESET        ;RESET DEFAULT PATTERN IN OUTPUT
                                                        ;BUFFER.
053716  077330                 SOB   R3,G34


                         ;TEST ACCUMULATOR 5 WITH FLOATING ONE
053720  012700  054252         MOV   #GPA.00,R0
053724  012701  054312         MOV   #GDAT00,R1
053730  004737  054124         JSR   PC,@#GSETUP        ;LOAD TEST PATTERN.
053734  012703  000102         MOV   #102,R3
053740                   G37:
053740  172410                 LDD   (R0),AC0
053742  174005                 STD   AC0,AC5
053744  172405                 LDD   AC5,AC0      ;STORE THE TEST PATTERN.
053746  174011                 STD   AC0,(R1)
053750  004737  054222         JSR   PC,@#GCMP          ;COMPARE THE DATA READ WITH
                                                        ;THAT WHICH WAS WRITTEN.
053754  005737  054246         TST   @#GFLAG1
053760  001004                 BNE   G40
053762  005137  054246         COM   @#GFLAG1
053766  000261                 SEC
053770  000401                 BR    G41
053772  000241           G40:  CLC
053774  006160  000006   G41:  ROL   6(R0)              ;GENERATE THE NEXT TEST PATTERN.
054000  006160  000004         ROL   4(R0)
054004  006160  000002         ROL   2(R0)
054010  006110                 ROL   (R0)
054012  004737  054202         JSR   PC,@#GRESET        ;RESET DEFAULT PATTERN IN OUTPUT
                                                        ;BUFFER.
054016  077330                 SOB   R3,G37


                         ;TEST ACCUMULATOR 5 WITH FLOATING ZERO
054020  012700  054262         MOV   #GPAT10,R0
054024  012701  054312         MOV   #GDAT00,R1
054030  004737  054124         JSR   PC,@#GSETUP        ;LOAD TEST PATTERN.
054034  012703  000102         MOV   #102,R3
054040                   G42:
054040  172410                 LDD   (R0),AC0
054042  174005                 STD   AC0,AC5
054044  172405                 LDD   AC5,AC0      ;STORE THE TEST PATTERN.
054046  174011                 STD   AC0,(R1)
054050  004737  054222         JSR   PC,@#GCMP          ;COMPARE THE DATA READ WITH
                                                        ;THAT WHICH WAS WRITTEN.
054054  005737  054246         TST   @#GFLAG1
054060  001004                 BNE   G43
054062  005137  054246         COM   @#GFLAG1
054066  000241                 CLC
054070  000401                 BR    G44
054072  000261           G43:  SEC
054074  006160  000006   G44:  ROL   6(R0)              ;GENERATE THE NEXT TEST PATTERN.
054100  006160  000004         ROL   4(R0)
054104  006160  000002         ROL   2(R0)
054110  006110                 ROL   (R0)
```

```
054112  004737  054202              JSR     PC,@#GRESET             ;RESET DEFAULT PATTERN IN OUTPUT
                                                                    ;BUFFER.
054116  077330                      SOB     R3,G42


054120  000137  054322              JMP     @#GDONE

                            ;USE THIS ROUTINE TO INITIALIZE ALL THE DATA BUFFERS.
054124  012705  054246      GSETUP: MOV     #GFLAG1,R5
054130  012704  000026              MOV     #26,R4
054134  005025              1$:     CLR     (R5)+
054136  077402                      SOB     R4,1$

054140  012705  054262              MOV     #GPAT10,R5
054144  012704  000010              MOV     #10,R4
054150  005125              2$:     COM     (R5)+
054152  077402                      SOB     R4,2$

054154  020067  000072      GS1:    CMP     R0,GPAT00
054160  001401                      BEQ     3$
054162  000207                      RTS     PC

054164  012705  054312      3$:     MOV     #GDAT00,R5
054170  012704  000004              MOV     #4,R4
054174  005125              4$:     COM     (R5)+
054176  077402                      SOB     R4,4$
054200  000207                      RTS     PC

054202  012705  054312      GRESET: MOV     #GDAT00,R5
054206  012704  000004              MOV     #4,R4
054212  005025              1$:     CLR     (R5.+
054214  077402                      SOB     R4,1$
054216  000137  054154              JMP     @#GS1

                            ;SEE IF THE DATA WRITTEN MATCHES THE DATA READ.
054222  012705  054312      GCMP:   MOV     #GDAT00,R5
054226  012704  000004              MOV     #4,R4
054232  010002                      MOV     R0,R2
054234  022225              1$:     CMP     (R2)+,(R5)+
054236  001401                      BEQ     2$
054240  104000                      EMT                     ;
054242  077404              2$:     SOB     R4,1$
054244  000207                      RTS     PC




054246  000000              GFLAG1: 0
054250  000000              GFLAG2: 0

054252  000000              GPAT00: 0
054254  000000              GPAT01: 0
054256  000000              GPAT02: 0
054260  000000              GPAT03: 0

054262  177777              GPAT10: -1
```

```
054264  177777             GPAT11: -1
054266  177777             GPAT12: -1
054270  177777             GPAT13: -1

054272  177777             GAND0:  -1
054274  177777             GAND1:  -1
054276  177777             GAND2:  -1
054300  177777             GAND3:  -1

054302  000000             GOR0:   0
054304  000000             GOR1:   0
054306  000000             GOR2:   0
054310  000000             GOR3:   0

054312  000000             GDAT00: 0
054314  000000             GDAT01: 0
054316  000000             GDAT02: 0
054320  000000             GDAT03: 0


054322                     GDONE:
054322  004767  044146             JSR     PC,.RSET        ;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?).



                           ;*******************************************************************
                           ;TEST 431       FPP ACCUMULATORS DUAL ADDRESS TEST
                           ;*******************************************************************

054326                     TS431:
054326  005037  055016     H1:     CLR     @#HFLAG
054332  012700  055020             MOV     #HA1W,R0                ;INITIALIZE THE LOAD BUFFER DATA.
054336  012701  055140             MOV     #HDAT1,R1
054342  012703  000024             MOV     #24,R3
054346  012120             H2:     MOV     (R1)+,(R0)+
054350  077302                     SOB     R3,H2

054352  004767  000420             JSR     PC,HCLR         ;CLEAR THE OUTPUT DATA BUFFER.

054356  170011             H3:     SETD
                           ;LOAD ACCUMULATOR 1
054360  012700  055020             MOV     #HA1W,R0
054364  172410                     LDD     (R0),AC0
054366  174001                     STD     AC0,AC1
                           ;LOAD ACCUMULATOR 2
054370  012700  055030             MOV     #HA2W,R0
054374  172410                     LDD     (R0),AC0
054376  174002                     STD     AC0,AC2
                           ;LOAD ACCUMULATOR 3
054400  012700  055040             MOV     #HA3W,R0
054404  172410                     LDD     (R0),AC0
054406  174003                     STD     AC0,AC3
                           ;LOAD ACCUMULATOR 4
054410  012700  055050             MOV     #HA4W,R0
```

```
054414  172410                  LDD     (R0),AC0
054416  174004                  STD     AC0,AC4

                        ;LOAD ACCUMULATOR 5
054420  012700  055060          MOV     #HA5W,R0
054424  172410                  LDD     (R0),AC0
054426  174005                  STD     AC0,AC5

054430  004737  054664  H4:     JSR     PC,@#HSTD               ;GO READ ALL ACCUMULATORS BACK.

054434  004737  054742          JSR     PC,@#HCMP               ;SEE IF DATA IS CORRECT.


                        ;COMPLIMENT EACH WORD OF THE DATA STORED IN ACCUMULATOR 1,
                        ;RELOAD THAT ACCUMULATOR, READ ALL THE ACCUMULATORS BACK AND CHECK
                        ;THE DATA.
054440  012700  055020          MOV     #HA1W,R0
054444  012702  000004          MOV     #4,R2
054450  010001                  MOV     R0,R1
054452  005121          H5:     COM     (R1)+
054454  172410                  LDD     (R0),AC0
054456  174001                  STD     AC0,AC1
054460  004737  054664          JSR     PC,@#HSTD               ;READ ALL THE ACCUMULATORS BACK.
054464  004737  054742          JSR     PC,@#HCMP               ;CHECK THE DATA.
054470  077210                  SOB     R2,H5

                        ;COMPLIMENT EACH WORD OF THE DATA STORED IN ACCUMULATOR 2,
                        ;RELOAD THAT ACCUMULATOR, READ ALL THE ACCUMULATORS BACK AND CHECK
                        ;THE DATA.
054472  012700  055030          MOV     #HA2W,R0
054476  012702  000004          MOV     #4,R2
054502  010001                  MOV     R0,R1
054504  005121          H6:     COM     (R1)+
054506  172410                  LDD     (R0),AC0
054510  174002                  STD     AC0,AC2
054512  004737  054664          JSR     PC,@#HSTD               ;READ ALL THE ACCUMULATORS BACK.
054516  004737  054742          JSR     PC,@#HCMP               ;CHECK THE DATA.
054522  077210                  SOB     R2,H6

                        ;COMPLIMENT EACH WORD OF THE DATA STORED IN ACCUMULATOR 3,
                        ;RELOAD THAT ACCUMULATOR, READ ALL THE ACCUMULATORS BACK AND CHECK
                        ;THE DATA.
054524  012700  055040          MOV     #HA3W,R0
054530  012702  000004          MOV     #4,R2
054534  010001                  MOV     R0,R1
054536  005121          H7:     COM     (R1)+
054540  172410                  LDD     (R0),AC0
054542  174003                  STD     AC0,AC3
054544  004737  054664          JSR     PC,@#HSTD               ;READ ALL THE ACCUMULATORS BACK.
054550  004737  054742          JSR     PC,@#HCMP               ;CHECK THE DATA.
054554  077210                  SOB     R2,H7

                        ;COMPLIMENT EACH WORD OF THE DATA STORED IN ACCUMULATOR 4,
                        ;RELOAD THAT ACCUMULATOR, READ ALL THE ACCUMULATORS BACK AND CHECK
                        ;THE DATA.
054556  012700  055050          MOV     #HA4W,R0
054562  012702  000004          MOV     #4,R2
054566  010001                  MOV     R0,R1
```

```
054570  005121                  H10:    COM     (R1)+
054572  172410                          LDD     (R0),AC0
054574  174004                          STD     AC0,AC4
054576  004737  054664                  JSR     PC,@#HSTD               ;READ ALL THE ACCUMULATORS BACK.
054602  004737  054742                  JSR     PC,@#HCMP               ;CHECK THE DATA.
054606  077210                          SOB     R2,H10

                                ;COMPLIMENT EACH WORD OF THE DATA STORED IN ACCUMULATOR 5,
                                ;RELOAD THAT ACCUMULATOR, READ ALL THE ACCUMULATORS BACK AND CHECK
                                ;THE DATA.
054610  012700  055060                  MOV     #HA5W,R0
054614  012702  000004                  MOV     #4,R2
054620  010001                          MOV     R0,R1
054622  005121                  H11:    COM     (R1)+
054624  172410                          LDD     (R0),AC0
054626  174005                          STD     AC0,AC5
054630  004737  054664                  JSR     PC,@#HSTD               ;READ ALL THE ACCUMULATORS BACK.
054634  004737  054742                  JSR     PC,@#HCMP               ;CHECK THE DATA.
054640  077210                          SOB     R2,H11


054642  005737  055016                  TST     @#HFLAG
054646  001402                          BEQ     H12
054650  000137  055210                  JMP     @#HDONE

054654  005137  055016          H12:    COM     @#HFLAG
054660  000137  054356                  JMP     @#H3

                                ;STORE ALL ACCUMULATORS IN THE OUTPUT BUFFERS.
054664  004737  054776          HSTD:   JSR     PC,@#HCLR               ;CLEAR ALL OUTPUT BUFFERS.
                                ;STORE ACCUMULATOR 1
054670  012704  055070                  MOV     #HA1R,R4
054674  172401                          LDD     AC1,AC0
054676  174014                          STD     AC0,(R4)
                                ;STORE ACCUMULATOR 2
054700  012704  055100                  MOV     #HA2R,R4
054704  172402                          LDD     AC2,AC0
054706  174014                          STD     AC0,(R4)
                                ;STORE ACCUMULATOR 3
054710  012704  055110                  MOV     #HA3R,R4
054714  172403                          LDD     AC3,AC0
054716  174014                          STD     AC0,(R4)
                                ;STORE ACCUMULATOR 4
054720  012704  055120                  MOV     #HA4R,R4
054724  172404                          LDD     AC4,AC0
054726  174014                          STD     AC0,(R4)
                                ;STORE ACCUMULATOR 5
054730  012704  055130                  MOV     #HA5R,R4
054734  172405                          LDD     AC5,AC0
054736  174014                          STD     AC0,(R4)
054740  000207                          RTS     PC

                                ;COMPARE DATA LOADED WITH DATA READ.
054742  012637  055014          HCMP:   MOV     (SP)+,@#HADR            ;SAVE RETURN ADDRESS.
054746  012703  055020                  MOV     #HA1W,R3
054752  012704  055070                  MOV     #HA1R,R4
```

```
054756  012705  000024                           MOV     #24,R5
054762  022324                          HCMP1:   CMP     (R3)+,(R4)+
054764  001401                                   BEQ     HCMP2
054766  104000                                   EMT                     ;
054770  077504                          HCMP2:   SOB     R5,HCMP1
054772  000177  000016                           JMP     @HADR

                                        ;CLEAR THE DATA OUTPUT BUFFER.
054776  012704  055070                  HCLR:    MOV     #HA1R,R4
055002  012705  000024                           MOV     #24,R5
055006  005024                          HCLR1:   CLR     (R4)+
055010  077502                                   SOB     R5,HCLR1
055012  000207                                   RTS     PC

055014  000000                          HADR:    0
055016  000000                          HFLAG:   0

055020  000000  000000  000000          HA1W:    .WORD   0,0,0,0
055026  000000
055030  000000  000000  000000          HA2W:    .WORD   0,0,0,0
055036  000000
055040  000000  000000  000000          HA3W:    .WORD   0,0,0,0
055046  000000
055050  000000  000000  000000          HA4W:    .WORD   0,0,0,0
055056  000000
055060  000000  000000  000000          HA5W:    .WORD   0,0,0,0
055066  000000

055070  000000  000000  000000          HA1R:    .WORD   0,0,0,0
055076  000000
055100  000000  000000  000000          HA2R:    .WORD   0,0,0,0
055106  000000
055110  000000  000000  000000          HA3R:    .WORD   0,0,0,0
055116  000000
055120  000000  000000  000000          HA4R:    .WORD   0,0,0,0
055126  000000
055130  000000  000000  000000          HA5R:    .WORD   0,0,0,0
055136  000000

055140  073567  073567  073567          HDAT1:   .WORD   73567,73567,73567,73567
055146  073567
055150  063146  063146  063146          HDAT2:   .WORD   63146,63146,63146,63146
055156  063146
055160  010421  010421  010421          HDAT3:   .WORD   10421,10421,10421,10421
055166  010421
055170  031463  031463  031463          HDAT4:   .WORD   31463,31463,31463,31463
055176  031463
055200  042104  042104  042104          HDAT5:   .WORD   42104,42104,42104,42104
055206  042104

055210                                  HDONE:
055210  004767  043260                           JSR     PC,.RSET        ;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?).
```

```
                                      ;*************************************************************************
                                      ;TEST 432        FSRC MODE 0 WITH ILLEGAL ACCUMULATOR TEST
                                      ;*************************************************************************
055214                                TS432:
055214  170011                                SETD                            ;SET FD
055216  012700  055476                        MOV     #SPAT10,R0              ;LOAD AC0
055222  172410                                LDD     (R0),AC0

055224  012737  055402  000244                MOV     #SERR0,@#FPVECT         ;USE OF THE NON-EXISTENT AC-
                                                                              ;CUMULATOR SHOULD RESULT IN
                                                                              ;A TRAP TO 244.
055232  012700  000001                        MOV     #1,R0                   ;A FAILURE IN THE FSRC FLOWS
                                                                              ;WILL RESULT IN AN ODD ADDRESS
055236  012737  055310  000004                MOV     #SERR1,@#ERRVECT        ;TRAP TO 4.
055244  005003                                CLR     R3

055246  172407                        SX2:    LDD     AC7,AC0
055250  170000                        SX3:    CFCC
055252  005203                                INC     R3
055254  005203                        SX4:    INC     R3

055256  012701  055506                        MOV     #SDAT00,R1              ;NO TRAP OCCURRED!!
055262  174011                                STD     AC0,(R1)                ;SEE IF AC0 WAS MODIFIED.

055264  012701  055506                        MOV     #SDAT00,R1
055270  012702  055476                        MOV     #SPAT10,R2
055274  012703  000004                        MOV     #4,R3
055300  022122                        SX5:    CMP     (R1)+,(R2)+
055302  001401                                BEQ     SX6
055304  104000                                EMT                     ;
055306  077304                        SX6:    SOB     R3,SX5
055310                                SERR1:
055310  104000                                EMT                     ;

                                      ;NOW TEST AC6.
055312  170011                        SX7:    SETD
055314  012700  055476                        MOV     #SPAT10,R0              ;LOAD AC0
055320  172410                                LDD     (R0),AC0
055322  012737  055452  000244                MOV     #SERR4,@#FPVECT
055330  012700  000001                        MOV     #1,R0
055334  005003                                CLR     R3

055336  172406                        SX8:    LDD     AC6,AC0
055340  170000                        SX9:    CFCC
055342  005203                                INC     R3
055344  005203                        SX10:   INC     R3

055346  012701  055506                        MOV     #SDAT00,R1
055352  174011                                STD     AC0,(R1)                ;NO TRAP! GET AC0.

055354  012701  055506                        MOV     #SDAT00,R1              ;WAS AC0 MODIFIED.
055360  012702  055476                        MOV     #SPAT10,R2
055364  012703  000004                        MOV     #4,R3
055370  022122                        SX11:   CMP     (R1)+,(R2)+
```

```
055372  001401                                   BEQ     SX12
055374  104000                                   EMT                     ;
055376  077304                          SX12:    SOB     R3,SX11
055400  104000                                   EMT                     ;

                                        ;TRAPPED TO 244.
055402  021627  055250                  SERR0:   CMP     (SP),#SX3               ;PC OF TRAP CORRECT?
055406  001401                                   BEQ     1$
055410  104000                                   EMT                     ;
055412  012737  055312  055472          1$:      MOV     #SX7,@#SADR
055420  022626                          SERR10:  CMP     (SP)+,(SP)+
055422  005004                                   CLR     R4
055424  170204                                   STFPS   R4                      ;IS FPS CORRECT?
055426  022704  100200                           CMP     #100200,R4
055432  001326                                   BNE     SERR1

055434  005004                                   CLR     R4
055436  170304                                   STST    R4                      ;IS FEC CORRECT?
055440  022704  000002                           CMP     #2,R4
055444  001321                                   BNE     SERR1
055446  000177  000020                           JMP     @SADR

055452  021627  055340                  SERR4:   CMP     (SP),#SX9
055456  001401                                   BEQ     1$
055460  104000                                   EMT                     ;
055462  012737  055516  055472          1$:      MOV     #SDONE,@#SADR
055470  000753                                   BR      SERR10

055472  000000                          SADR:    0
055474  177777                                   -1
055476  010421                          SPAT10:  10421
055500  021042                          SPAT11:  21042
055502  031463                          SPAT12:  31463
055504  042104                          SPAT13:  42104

055506  000000                          SDAT00:  0
055510  000000                          SDAT01:  0
055512  000000                          SDAT02:  0
055514  000000                          SDAT03:  0

055516                                  SDONE:
055516  004767  042752                           JSR     PC,.RSET        ;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?).

                                        ;**************************************************************************
                                        ;TEST 433        FSRC MODE 2 TEST
                                        ;**************************************************************************
055522                                  TS433:
055522                                  J1:
055522  170011                                   SETD                    ;SET DOUBLE MODE

055524  012700  055652                           MOV     #JDAT0,R0
055530  172410                                   LDD     (R0),AC0        ;LOAD AC0=ALL 1
```

```

055532  012700  055632                  MOV     #JDATI0,R0
055536  005003                          CLR     R3

055540  172420                  J2:     LDD     (R0)+,AC0               ;TEST INSTRUCTION
055542  005203                  J3:     INC     R3
055544  005203                  J4:     INC     R3

055546  012701  055642                  MOV     #JDAT00,R1
055552  174011                          STD     AC0,(R1)                ;PICK UP RESULTS

055554  020027  055622                  CMP     R0,#JBUF0               ;WAS AN AUTO
055560  001001                          BNE     1$
055562  104000                          EMT                             ;
055564  012702  055632          1$:     MOV     #JDATI0,R2              ;IS DATA CORRECT?
055570  012703  055642                  MOV     #JDAT00,R3
055574  012704  000004                  MOV     #4,R4
055600  022223                  J5:     CMP     (R2)+,(R3)+
055602  001401                          BEQ     J6
055604  104000                          EMT                             ;
055606  077404                  J6:     SOB     R4,J5

055610  022700  055642                  CMP     #JDATI0+10,R0           ;WAS R0 INCREM.
055614  001401                          BEQ     J7
055616  104000                          EMT                             ;
055620  000420                  J7:     BR      JDONE

055622  010421                  JBUF0:  .WORD   010421
055624  021042                  JBUF1:          021042
055626  042104                  JBUF2:          042104
055630  031463                  JBUF3:          031463

055632  052525                  JDATI0:         052525
055634  114631                  JDATI1:         114631
055636  063146                  JDATI2:         063146
055640  073567                  JDATI3:         073567

055642  000000                  JDAT00:         0
055644  000000                  JDAT01:         0
055646  000000                  JDAT02:         0
055650  000000                  JDAT03:         0

055652  177777                  JDAT0:          -1
055654  177777                  JDAT1:          -1
055656  177777                  JDAT2:          -1
055660  177777                  JDAT3:          -1


055662                          JDONE:
055662  004767  042606                  JSR     PC,.RSET                ;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?).


```

```
                              ;*****************************************************************************
                              ;TEST 434        FSRC MODE 4 TEST
                              ;*****************************************************************************
055666                        TS434:
055666  170011                        SETD                          ;SET DOUBLE MODE
055670  012700  056016                MOV     #KPATO,R0
055674  172410                        LDD     (R0),AC0              ;LOAD A DEFAULT
                                                                    ;PATTERN INTO AC0
055676  012700  055776                MOV     #KBUF0,R0
055702  005003                        CLR     R3
055704  172440                KX2:    LDD     -(R0),AC0             ;TEST INSTRUCTION
055706  005203                KX3:    INC     R3
055710  005203                KX4:    INC     R3
055712  012701  056006                MOV     #KDAT00,R1
055716  174011                        STD     AC0,(R1)              ;PICK UP THE RESULT
055720  020027  056006                CMP     R0,#KBUF0+10          ;WAS AN AUTO
055724  001001                        BNE     1$
055726  104000                        EMT                           ;
055730  012702  055766        1$:     MOV     #KDATI0,R2            ;IS DATA CORRECT?
055734  012703  056006                MOV     #KDAT00,R3
055740  012704  000004                MOV     #4,R4
055744  022223                KX5:    CMP     (R2)+,(R3)+
055746  001401                        BEQ     KX6
055750  104000                        EMT                           ;
055752  077404                KX6:    SOB     R4,KX5
055754  022700  055766                CMP     #KBUF0-10,R0          ;WAS R0 DECREMENTED
055760  001401                        BEQ     KX7
055762  104000                        EMT                           ;
055764  000420                KX7:    BR      KDONE
055766  052525                KDATI0: .WORD   052525
055770  114631                KDATI1:         114631
055772  063140                KDATI2:         063140
055774  073567                KDATI3:         073567
055776  010421                KBUF0:          010421
056000  031463                KBUF1:          031463
056002  042104                KBUF2:          042104
056004  021042                KBUF3:          021042
056006  000000                KDAT00:         0
056010  000000                KDAT01:         0
056012  000000                KDAT02:         0
056014  000000                KDAT03:         0
056016  177777                KPAT0:          -1
056020  177777                KPAT1:          -1
056022  177777                KPAT2:          -1
056024  177777                DPAT3:          -1
056026                        KDONE:
056026  004767  042442                JSR     PC,.RSET              ;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?).


                                        ;*************************************************************************
                                        ;TEST 435        FSRC MODE 2, WITH FD=0, TEST
                                        ;*************************************************************************
056032                                  TS435:
056032                                  L1:
056032  170011                                  SETD                            ;SET DOUBLE MODE

056034  012700  056160                          MOV     #LPAT10,R0
056040  172410                                  LDD     (R0),AC0                ;LOAD AC0

056042  012700  056202                          MOV     #LDATI0,R0              ;SET UP THE INPUT
056046  012701  056170                          MOV     #LPAT20,R1              ;DATA
056052  012702  000004                          MOV     #4,R2

056056  012120                          1$:     MOV     (R1)+,(R0)+
056060  077202                                  SOB     R2,1$

056062  012700  056202                          MOV     #LDATI0,R0
056066  005003                                  CLR     R3
056070  170001                                  SETF                            ;CLEAR FD.

056072  172420                          L2:     LDF     (R0)+,AC0
056074  005203                          L3:     INC     R3

056076                                  L4:
056076  170011                                  SETD                            ;SET FD

056100  012701  056214                          MOV     #LDAT00,R1
056104  174011                                  STD     AC0,(R1)                ;PICK UP RESULTS

056106  020027  056206                          CMP     R0,#LDATI2              ;WAS R0 INCREMENTED
056112  001401                                  BEQ     1$
056114  104000                                  EMT                             ;
056116  012737  177777  056206          1$:     MOV     #-1,@#LDATI2
056124  012737  177777  056210                  MOV     #-1,@#LDATI3
056132  012702  056202                          MOV     #LDATI0,R2              ;IS DATA CORRECT
056136  012703  056214                          MOV     #LDAT00,R3
056142  012704  000004                          MOV     #4,R4

056146  022223                          L5:     CMP     (R2)+,(R3)+
056150  001401                                  BEQ     L6
056152  104000                                  EMT                             ;
056154  077404                          L6:     SOB     R4,L5
056156  000422                                  BR      LDONE

056160  177777                          LPAT10: .WORD   -1
056162  177777                          LPAT11:         -1
056164  177777                          LPAT12:         -1
056166  177777                          LPAT13:         -1
```

```
056170  052525                        LPAT20:          052525
056172  114631                        LPAT21:          114631
056174  063142                        LPAT22:          063142
056176  073567                        LPAT23:          073567
056200  000001                                 .WORD   000001
056202  000000                        LDATI0:          0
056204  000000                        LDATI1:          0
056206  000000                        LDATI2:          0
056210  000000                        LDATI3:          0
056212  000001                                 .WORD   00001
056214  000000                        LDATO0:          0
056216  000000                        LDATO1:          0
056220  000000                        LDATO2:          0
056222  000000                        LDATO3:          0

056224                                LDONE:
056224  004767  042244                         JSR     PC,.RSET          ;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?).


                                      ;*************************************************************************
                                      ;TEST 436        FSRC MODE 2 WITH GR7, IMMEDIATE MODE, TEST
                                      ;*************************************************************************
056230                                TS436:

056230                                M1:
056230  170011                                 SETD

056232  012700  056332                         MOV     #MPAT10,R0
056236  172410                                 LDD     (R0),AC0          ;LOAD BACKROUND
                                                                         ;PATTERN INTO AC0.
056240  005004                                 CLR     R4
056242  012737  056270  000004                 MOV     #MERR3,@#ERRVECT

056250  172427  000000                M15:     LDD     #0,AC0            ;TEST INSTRUCTION
        056252                                 .=.-2           ;EFFECTIVELY: 05204 IS PUT IN THE FIRST
056252  005204                                 .WORD   5204    ;16 BIT WORD, OR THE "EXP-FRACTION" WORD.
056254  005204                        M2:      INC     R4                ;NOTE THAT
056256  005204                        M3:      INC     R4                ;005204=INC R4
056260  005204                        M4:      INC     R4
056262  020427  000003                         CMP     R4,#3             ;SEE IF THE PC
056266  001401                                 BEQ     M8
056270                                MERR3:
056270  104000                                 EMT                       ;
056272  012700  056352                M8:      MOV     #MDAT00,R0
056276  174010                                 STD     AC0,(R0)          ;GET THE DATA

056300  012700  056352                         MOV     #MDAT00,R0
056304  022720  005204                         CMP     #5204,(R0)+       ;IS THE DATA CORRECT?
056310  001401                                 BEQ     M5
056312  104000                                 EMT                       ;
056314  012701  000003                M5:      MOV     #3,R1
056320  005720                        M6:      TST     (R0)+
```

```
056322  001401                          BEQ     M7
056324  104000                          EMT                             ;
056326  077104                  M7:     SOB     R1,M6
056330  000414                          BR      MDONE

056332  177777                  MPAT10:         -1
056334  177777                  MPAT11:         -1
056336  177777                  MPAT12:         -1
056340  177777                  MPAT13:         -1

056342  005204                  MPAT20:         5204
056344  005204                  MPAT21:         5204
056346  005204                  MPAT22:         5204
056350  005204                  MPAT23:         5204

056352  000000                  MDAT00:         0
056354  000000                  MDAT01:         0
056356  000000                  MDAT02:         0
056360  000000                  MDAT03:         0

056362                          MDONE:
056362  004767  042106                  JSR     PC,.RSET        ;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?).


                                ;*****************************************************************************
                                ;TEST 437       FSRC MODE 3 TEST
                                ;*****************************************************************************
056366                          TS437:

056366                          N1:
056366  170011                          SETD                    ;SET FD MODE

056370  012700  056516                  MOV     #NPAT10,R0
056374  172410                          LDD     (R0),AC0        ;LOAD AC0 WITH A DEFAULT
                                                                ;PATTERN
056376  012700  056504                  MOV     #NPAT20,R0
056402  005003                          CLR     R3
056404  012737  056456  000004          MOV     #NERRO,@#ERRVECT        ;IF A FAILURE OCCURS
                                                                ;IN THE FSRC FLOWS AN
                                                                ;ODD TRAP TO 4 COULD OCCUR
056412  172430                  N2:     LDD     @(R0)+,AC0      ;TEST INSTRUCTION.
056414  005203                  N3:     INC     R3
056416  005203                  N4:     INC     R3

056420  012701  056464                  MOV     #NDAT00,R1
056424  174011                          STD     AC0,(R1)        ;GET THE DATA

056426  020027  056506                  CMP     R0,#NPAT20+2    ;WAS R0 INCREMENTED
056432  001401                          BEQ     N12
056434  104000                          EMT                     ;
056436  012702  056464          N12:    MOV     #NDAT00,R2      ;DATA CORRECT
056442  012703  056526                  MOV     #NDAT10,R3
```

```
056446  012704  000004                  MOV     #4,R4
056452  022223                  N13:    CMP     (R2)+,(R3)+
056454  001401                          BEQ     N14
056456                          NERRO:
056456  104000                          EMT                     ;
056460  077404                  N14:    SOB     R4,N13
056462  000425                          BR      NDONE

056464  000000                  NDAT00: .WORD   0
056466  000000                  NDAT01:         0
056470  000000                  NDAT02:         0
056472  000000                  NDAT03:         0

056474  052525  052525  052525          .WORD   52525,52525,52525,52525
056502  052525
056504  056526                  NPAT20: .WORD   NDAT10
056506  070707                  NPAT21:         070707
056510  070707                  NPAT22:         070707
056512  070707                  NPAT23:         070707
056514  000001                          .WORD   1
056516  177777                  NPAT10: .WORD   -1
056520  177777                  NPAT11:         -1
056522  177777                  NPAT12:         -1
056524  177777                  NPAT13:         -1

056526  010421                  NDAT10: .WORD   010421
056530  021042                  NDAT11:         021042
056532  031463                  NDAT12:         031463
056534  042104                  NDAT13:         042104

056536                          NDONE:
056536  004767  041732                  JSR     PC,.RSET        ;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?).


                                ;*******************************************************************************
                                ;TEST 440       FSRC MODE 5 TEST
                                ;*******************************************************************************
056542                          TS440:

056542                          01:
056542  170011                          SETD                    ;SET FD MODE

056544  012700  056672                  MOV     #OPAT10,R0
056550  172410                          LDD     (R0),AC0        ;LOAD AC0 WITH A
                                                                ;DEFAULT PATTERN.
056552  012700  056660                  MOV     #OPAT21,R0
056556  005003                          CLR     R3
056560  012737  056610  000004          MOV     #OERRO,@#ERRVEC ;IF A FAILURE
                                                                ;OCCURS IN THE FSRC
                                                                ;FLOWS AN ODD ADDR.
                                                                ;TRAP TO 4 MAY OCCUR.
056566  172450                  02:     LDD     @-(R0),AC0      ;TEST INSTRUCTION
```

```
056570  005203                   03:     INC     R3
056572  005203                   04:     INC     R3

056574  012701  056640                   MOV     #ODAT00,R1
056600  174011                           STD     AC0,(R1)          ;GET THE DATA

056602  020027  056656                   CMP     R0,#OPAT20        ;WAS R0 DECREMENTED
056606  001401                           BEQ     012
056610                           OERR0:
056610  104000                           EMT                       ;
056612  012702  056640           012:    MOV     #ODAT00,R2        ;DATA CORRECT?
056616  012703  056702                   MOV     #ODAT10,R3
056622  012704  000004                   MOV     #4,R4
056626  022223                   013:    CMP     (R2)+,(R3)+
056630  001401                           BEQ     014
056632  104000                           EMT                       ;
056634  077404                   014:    SOB     R4,013
056636  000425                           BR      ODONE

056640  000000                   ODAT00: .WORD   0
056642  000000                   ODAT01:         0
056644  000000                   ODAT02:         0
056646  000000                   ODAT03:         0

056650  052525  052525  052525           .WORD   52525,52525,52525
056656  056702                   OPAT20: .WORD   ODAT10
056660  070707                   OPAT21: 070707
056662  070707                   OPAT22: 070707
056664  070707                   OPAT23: 070707
056666  070707                   OPAT24: 070707
056670  000001                           .WORD   1
056672  177777                   OPAT10: .WORD   -1
056674  177777                   OPAT11:         -1
056676  177777                   OPAT12:         -1
056700  177777                   OPAT13:         -1

056702  073567                   ODAT10: .WORD   73567
056704  004210                   ODAT11:         004210
056706  114631                   ODAT12:         114631
056710  125252                   ODAT13:         125252

056712                           ODONE:
056712  004767  041556                   JSR     PC,.RSET          ;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?).


                                 ;*****************************************************************
                                 ;TEST 441       FSRC MODE 6 TEST
                                 ;*****************************************************************
056716                           TS441:

056716                           P1:
056716  170011                           SETD                      ;SET FD MODE
```

```

056720  012700  057004                  MOV     #PPAT10,R0
056724  172410                          LDD     (R0),AC0        ;LOAD A DEFAULT PATTERN
                                                                ;INTO AC0
056726  012700  056553                  MOV     #PDATI0-241,R0  ;COULD OCCUR.

056732  172460  000241          P2:     LDD     241(R0),AC0
        056734                  P3=P2+2

056736  012701  057024          P4:     MOV     #PDAT00,R1
056742  174011                          STD     AC0,(R1)        ;GET THE DATA
056744  012703  000004                  MOV     #4,R3
056750  012702  057014                  MOV     #PDATI0,R2
056754  012701  057024                  MOV     #PDAT00,R1
056760  022221                  P5:     CMP     (R2)+,(R1)+     ;CHECK THE DATA
056762  001401                          BEQ     2$
056764  104000                          EMT                     ;
056766  077304                  2$:     SOB     R3,P5
056770  022700  056553                  CMP     #PDATI0-241,R0  ;R0 CORRECT?
056774  001401                          BEQ     1$
056776  104000                          EMT                     ;
057000  000137  057034          1$:     JMP     @#PDONE
057004  177777                  PPAT10: .WORD   -1
057006  177777                  PPAT11:         -1
057010  177777                  PPAT12:         -1
057012  177777                  PPAT13:         -1

057014  010421                  PDATI0: .WORD   010421
057016  031463                  PDATI1:         031463
057020  052525                  PDATI2:         052525
057022  073567                  PDATI3:         073567

057024  000000                  PDAT00: .WORD   0
057026  000000                  PDAT01:         0
057030  000000                  PDAT02:         0
057032  000000                  PDAT03:         0

057034                          PDONE:
057034  004767  041434                  JSR     PC,.RSET        ;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?).



                                ;*********************************************************************
                                ;TEST 442       FSRC MODE 7 TEST
                                ;*********************************************************************
057040                          TS442:

057040                          Q1:
057040  170011                          SETD

057042  012700  057126                  MOV     #QPAT10,R0
057046  172410                          LDD     (R0),AC0        ;LOAD A DEFAULT
                                                                ;PATTERN INTO AC0

```

```
057050  012700  056675                    MOV     #QPAT20-241,R0

057054  172470  000241          Q2:       LDD     @241(R0),AC0
        057056                  Q3=Q2+2

057060  012701  057146          Q4:       MOV     #QDAT00,R1
057064  174011                            STD     AC0,(R1)        ;GET THE DATA

057066  012703  000004                    MOV     #4,R3
057072  012704  057146                    MOV     #QDAT00,R4
057076  012705  057156                    MOV     #QDAT10,R5
057102  022425                  Q5:       CMP     (R4)+,(R5)+     ;CHECK THE DATA
057104  001401                            BEQ     2$
057106  104000                            EMT                     ;
057110  077304                  2$:       SOB     R3,Q5

057112  022700  056675                    CMP     #QPAT20-241,R0  ;CHECK R0.
057116  001401                            BEQ     1$
057120  104000                            EMT                     ;
057122  000137  057166          1$:       JMP     @#QDONE

057126  177777                  QPAT10:   .WORD   -1
057130  177777                  QPAT11:           -1
057132  177777                  QPAT12:           -1
057134  177777                  QPAT13:           -1

057136  057156                  QPAT20:   .WORD   QDAT10
057140  052525                  QPAT21:           52525
057142  052525                  QPAT22:           52525
057144  052525                  QPAT23:           52525

057146  000000                  QDAT00:   .WORD   0
057150  000000                  QDAT01:           0
057152  000000                  QDAT02:           0
057154  000000                  QDAT03:           0

057156  073567                  QDAT10:   .WORD   073567
057160  052525                  QDAT11:   .WORD   052525
057162  031463                  QDAT12:   .WORD   031463
057164  010421                  QDAT13:   .WORD   010421

057166                          QDONE:
057166  004767  041302                    JSR     PC,.RSET        ;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?).

                                ;*********************************************************************
                                ;TEST 443         (BUT EZBT Y8),(BUT ENBT) AND (BUT FIUV) TEST
                                ;*********************************************************************
057172                          TS443:
057172  005037  057744                    CLR     @#UFLAG
057176  012700  057674                    MOV     #UPAT00,R0      ;SET UP AC#0 DATA.
057202  012701  000004                    MOV     #4,R1
057206  012720  177777          U0:       MOV     #-1,(R0)+
```

```
057212  077103                           SOB     R1,U0

057214  012737  000033  057746           MOV     #033,@#UTMP1
057222  012737  000023  057750           MOV     #023,@#UTMP2
057230  012737  057314  000244   U1:     MOV     #UERRO,@#FPVECT ;IN CASE (BUT FIUV FAILS)
057236  012700  000200                   MOV     #200,R0
057242  170100                           LDFPS   R0
057244  012700  057674                   MOV     #UPAT00,R0      ;LOAD AC0
057250  172410                           LDD     (R0),AC0
057252  013737  057746  057752           MOV     @#UTMP1,@#UROM1
057260  012737  000001  057754           MOV     #001,@#UROM2
057266  012737  000254  057756           MOV     #254,@#UROM3

057274  012700  057704                   MOV     #UPAT10,R0      ;LOAD 0 INTO AC0
057300  172410                   U2:     LDD     (R0),AC0
057302  012704  000204                   MOV     #204,R4         ;SEE IF FPS IS CORRECT
057306  170205                           STFPS   R5

057310  020405                           CMP     R4,R5
057312  001401                           BEQ     U3
057314                           UERRO:
057314  104000                           EMT                     ;
057316  012700  000200           U3:     MOV     #200,R0
057322  170100                           LDFPS   R0

057324  012700  057674                   MOV     #UPAT00,R0      ;LOAD AC0
057330  172410                           LDD     (R0),AC0
057332  013737  057750  057752           MOV     @#UTMP2,@#UROM1
057340  012737  000003  057754           MOV     #003,@#UROM2
057346  012737  000054  057756           MOV     #054,@#UROM3

057354  012700  057714                   MOV     #UPAT20,R0      ;LOAD A POSITIVE NUMBER
                                                                 ;INTO AC0

057360  172410                   U4:     LDD     (R0),AC0
057362  012704  000200                   MOV     #200,R4         ;FPS CORRECT?
057366  170205                           STFPS   R5

057370  020405                           CMP     R4,R5
057372  001401                           BEQ     U5
057374  104000                           EMT                     ;
057376  012700  000200           U5:     MOV     #200,R0
057402  170100                           LDFPS   R0
057404  012700  057674                   MOV     #UPAT00,R0      ;LOAD AC0
057410  172410                           LDD     (R0),AC0
057412  013737  057750  057752           MOV     @#UTMP2,@#UROM1
057420  012737  000403  057754           MOV     #403,@#UROM2
057426  012737  000056  057756           MOV     #056,@#UROM3
057434  012700  057724                   MOV     #UPAT30,R0      ;LOAD A NEGATIUE
                                                                 ;NUMBER INTO ALO
057440  172410                   U6:     LDD     (R0),AC0
057442  012704  000210                   MOV     #210,R4         ;FPS CORRECT
057446  170205                           STFPS   R5
057450  020405                           CMP     R4,R5
057452  001401                           BEQ     U7
057454  104000                           EMT                     ;
057456  012700  000200           U7:     MOV     #200,R0
```

```
057462  170100                                  LDFPS   R0
057464  012700  057674                          MOV     #UPAT00,R0      ;LOAD AC0
057470  172410                                  LDD     (R0),AC0
057472  013737  057746  057752                  MOV     @#UTMP1,@#UROM1
057500  012737  000401  057754                  MOV     #401,@#UROM2
057506  012737  000256  057756                  MOV     #256,@#UROM3
057514  012700  057734                          MOV     #UPAT40,R0      ;LOAD -0 INTO AC0
057520  172410                          U10:    LDD     (R0),AC0
057522  000240                          U11:    NOP                     ;TRAP FROM HERE IF
057524  012704  000214                          MOV     #214,R4         ;SEE IF FPS IS CORRECT.
057530  170205                                  STFPS   R5
057532  020405                                  CMP     R4,R5
057534  001401                                  BEQ     U12
057536  104000                                  EMT                     ;
057540  005737  057744                  U12:    TST     @#UFLAG ;SEE IF ALL THE PATTERNS
057544  001021                                  BNE     U14             ;HAVE BEEN TEST WITH
                                                                        ;BOTH AC NOT EQUAL TO 0 AND AC=0
057546  012700  057674                          MOV     #UPAT00,R0      ;IF NOT GO BACK AND
057552  012701  000004                          MOV     #4,R1           ;CHECK THEM WITH AC=0
057556  005020                          U13:    CLR     (R0)+
057560  077102                                  SOB     R1,U13
057562  012737  177777  057744                  MOV     #-1,@#UFLAG
057570  012737  000233  057746                  MOV     #233,@#UTMP1
057576  012737  000223  057750                  MOV     #223,@#UTMP2
057604  000137  057230                          JMP     @#U1
                                        ;NOW SEE IF A TRAP CAN BE FORCED BY SETTING FIUV AND LOADING -0
057610  012737  057646  000244          U14:    MOV     #UERR3,@#FPVECT
057616  012700  004200                          MOV     #4200,R0        ;SET FD AND FIUV
057622  170100                                  LDFPS   R0
057624  012700  057674                          MOV     #UPAT00,R0      ;SET UP AC0
057630  172410                                  LDD     (R0),AC0
057632  012700  057734                          MOV     #UPAT40,R0      ;LOAD -0
057636  172410                          U15:    LDD     (R0),AC0        ;SHOULD TRAP TO 244
057640  170000                          U16:    CFCC
057642  000240                                  NOP
057644  104000                                  EMT                     ;

                                        ;INTERRUPT HERE WHEN FIUV SET AND ATTEMPTED TO LOAD-0
057646  021627  057640                  UERR3:  CMP     (SP),#U16
057652  001401                                  BEQ     1$
057654  104000                                  EMT                     ;
057656  022626                          1$:     CMP     (SP)+,(SP)+
057660  005000                                  CLR     R0
057662  170300                                  STST    R0              ;GET FEC.
057664  022700  000014                          CMP     #14,R0          ;CORRECT
057670  001433                                  BEQ     UDONE
057672  104000                                  EMT                     ;
057674  000000                          UPAT00: .WORD   0
057676  000000                          UPAT01:         0
057700  000000                          UPAT02:         0
057702  000000                          UPAT03:         0

057704  000000                          UPAT10: .WORD   0               ;0
057706  000000                          UPAT11:         0
057710  000000                          UPAT12:         0
057712  000000                          UPAT13:         0
```

```

057714  010421                  UPAT20: .WORD   010421          ;POS NUM
057716  114631                  UPAT21:         114631
057720  125252                  UPAT22:         125252
057722  177777                  UPAT23:         177777

057724  114631                  UPAT30:         114631          ;NEG NUM
057726  135673                  UPAT31:         135673
057730  146314                  UPAT32:         146314
057732  167356                  UPAT33:         167356

057734  100000                  UPAT40:         100000          ;NEG ZERO
057736  000000                  UPAT41:         0
057740  000000                  UPAT42:         0
057742  000000                  UPAT43:         0

057744  000000                  UFLAG:  .WORD   0
057746  000000                  UTMP1:          0
057750  000000                  UTMP2:          0
057752  000000                  UROM1:          0
057754  000000                  UROM2:          0
057756  000000                  UROM3:          0
057760                          UDONE:



                                ;*************************************************************************
                                ;TEST 444       ADDF,ADDD,SUBF AND SUBD WITH FSRC=AC=0 TEST
                                ;*************************************************************************
057760                          TS444:
057760  012700  000200                  MOV     #200,R0
057764  170100                          LDFPS   R0              ;SET DOUBLE MODE
057766  012700  060310                  MOV     #WPAT00,R0      ;LOAD AC0=:
057772  172410                          LDD     (R0),AC0
057774  012700  060310                  MOV     #WPAT00,R0
060000  172010                  W2:     ADDD    (R0),AC0        ;TEST INSTRUCTION. ADD ITSELF
060002  170205                          STFPS   R5              ;GET FPS
060004  170011                          SETD                    ;SET DOUBLE MODE
060006  012700  060310                  MOV     #WPAT00,R0
060012  174010                          STD     AC0,(R0)        ;GET THE RESULT
060014  012701  060310                  MOV     #WPAT00,R1
060020  012702  000004                  MOV     #4,R2
060024  022021                  W3:     CMP     (R0)+,(R1)+     ;IS RESULT CORRECT
060026  001401                          BEQ     W4
060030  104000                          EMT                     ;
060032  077204                  W4:     SOB     R2,W3
060034  022705  000204                  CMP     #204,R5         ;IS FPS CORRECT
060040  001401                          BEQ     W5
060042  104000                          EMT                     ;
060044  012700  000200          W5:     MOV     #200,R0
060050  170100                          LDFPS   R0              ;SET DOUBLE MODE
060052  012700  060310                  MOV     #WPAT00,R0      ;LOAD AC0=0
060056  172410                          LDD     (R0),AC0
060060  005000                          CLR     R0
060062  170100                          LDFPS   R0              ;GO TO FLOATING MODE
060064  012700  060310                  MOV     #WPAT00,R0
060070  172010                  W6:     ADDF    (R0),AC0        ;TEST INSTRUCTION
```

```
060072  170205                   STFPS   R5              ;GET FPS
060074  170011                   SETD                    ;RESET TO DOUBLE MODE
060076  012700  060310           MOV     #WPAT00,R0
060102  174010                   STD     AC0,(R0)        ;GET THE RESULT
060104  012701  060310           MOV     #WPAT00,R1
060110  012702  000004           MOV     #4,R2
060114  022021           W7:     CMP     (R0)+,(R1)+     ;WAS THE RESULT
060116  001401                   BEQ     W10
060120  104000                   EMT                     ;
060122  077204           W10:    SOB     R2,W7
060124  022705  000004           CMP     #4,R5           ;WAS FPS CORRECT
060130  001401                   BEQ     W11
060132  104000                   EMT                     ;
060134  012700  000200   W11:    MOV     #200,R0
060140  170100                   LDFPS   R0              ;SET DOUBLE MODE
060142  012700  060310           MOV     #WPAT00,R0      ;LOAD AC0=0
060146  172410                   LDD     (R0),AC0
060150  012700  060310           MOV     #WPAT00,R0
060154  173010           W12:    SUBD    (R0),AC0        ;TEST INSTRUCTION
060156  170205                   STFPS   R5              ;GET FPS
060160  170011                   SETD                    ;SET DOUBLE MODE
060162  012700  060310           MOV     #WPAT00,R0
060166  174010                   STD     AC0,(R0)        ;GET THE RESULT
060170  012701  060310           MOV     #WPAT00,R1
060174  012702  000004           MOV     #4,R2
060200  022021           W13:    CMP     (R0)+,(R1)+     ;IS RESULT CORRECT?
060202  001401                   BEQ     W14
060204  104000                   EMT                     ;
060206  077204           W14:    SOB     R2,W13
060210  022705  000204           CMP     #204,R5         ;IS FPS CORRECT?
060214  001401                   BEQ     W15
060216  104000                   EMT                     ;
060220  012700  000200   W15:    MOV     #200,R0
060224  170100                   LDFPS   R0              ;SET DOUBLE MODE
060226  012700  060310           MOV     #WPAT00,R0      ;LOAD AC0=0
060232  172410                   LDD     (R0),AC0
060234  005000                   CLR     R0
060236  170100                   LDFPS   R0              ;ENTER FLOATING MODE.
060240  012700  060310           MOV     #WPAT00,R0
060244  173010           W16:    SUBF    (R0),AC0        ;TEST INSTRUCTION.
060246  170205                   STFPS   R5              ;GET FPS
060250  170011                   SETD                    ;RESET TO DOUBLE MODE
060252  012700  060310           MOV     #WPAT00,R0      ;GET THE RESULT.
060256  174010                   STD     AC0,(R0)
060260  012701  060310           MOV     #WPAT00,R1
060264  012702  000004           MOV     #4,R2
060270  022021           W17:    CMP     (R0)+,(R1)+     ;IS RESULT CORRECT?
060272  001401                   BEQ     W20
060274  104000                   EMT                     ;
060276  077204           W20:    SOB     R2,W17
060300  022705  000004           CMP     #4,R5           ;IS FPS CORRECT?
060304  001411                   BEQ     WDONE
060306  104000                   EMT                     ;

060310  000000           WPAT00: .WORD   0
060312  000000           WPAT01:         0
```

```
060314  000000                  WPAT02:          0
060316  000000                  WPAT03:          0

060320  000000                  WDAP00: .WORD    0
060322  000000                  WDAT01:          0
060324  000000                  WDAT02:          0
060326  000000                  WDAT03:          0

060330                          WDONE:
060330  004767  040140                  JSR      PC,.RSET          ;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?).


                                ;*************************************************************************
                                ;TEST 445        ADDD AND SUB WITH FSRC=0
                                ;*************************************************************************
060334                          TS445:
060334  012700  000200                  MOV      #200,R0
060340  170100                          LDFPS    R0                ;SET DOUBLE MODE
060342  012700  060674                  MOV      #XPAT00,R0        ;SET AC0 TO POSITIVE
060346  172410                          LDD      (R0),AC0
060350  012700  060704                  MOV      #XPAT10,R0        ;FSRC=0
060354  172010                  X2:     ADDD     (R0),AC0          ;TEST INSTRUCTION
060356  170205                          STFPS    R5
060360  170011                          SETD
060362  012700  060664                  MOV      #XDAT00,R0        ;GET RESULT.
060366  174010                          STD      AC0,(R0)
060370  012701  060674                  MOV      #XPAT00,R1
060374  012702  000004                  MOV      #4,R2
060400  022021                  X3.     CMP      (R0)+,(R1)+       ;IS RESULT CORRECT?
060402  001401                          BEQ      X4
060404  104000                          EMT                        ;
060406  077204                  X4:     SOB      R2,X3
060410  012704  000200                  MOV      #200,R4
060414  020405                          CMP      R4,R5             ;IS FPS CORRECT?
060416  001401                          BEQ      X5
060420  104000                          EMT                        ;
060422  012700  000200          X5:     MOV      #200,R0
060426  170100                          LDFPS    R0                ;SET DOUBLE MODE
060430  012700  060714                  MOV      #XPAT20,R0        ;SET AC0 TO
060434  172410                          LDD      (R0),AC0
060436  012700  060704                  MOV      #XPAT10,R0        ;FSRC=0
060442  172010                  X6:     ADDD     (R0),AC0          ;TEST INSTRUCTION
060444  170205                          STFPS    R5
060446  170011                          SETD
060450  012700  060664                  MOV      #XDAT00,R0        ;GET RESULT
060454  174010                          STD      AC0,(R0)
060456  012701  060714                  MOV      #XPAT20,R1
060462  012702  000004                  MOV      #4,R2
060466  022021                  X7:     CMP      (R0)+,(R1)+       ;IS RESULT CORRECT?
060470  001401                          BEQ      X10
060472  104000                          EMT                        ;
060474  077204                  X10:    SOB      R2,X7
```

```
060476 012704 000210            MOV     #210,R4
060502 020405                   CMP     R4,R5           ;IS FPS CORRECT?
060504 001401                   BEQ     X11
060506 104000                   EMT                     ;
060510 012700 000200    X11:    MOV     #200,R0
060514 170100                   LDFPS   R0              ;SET DOUBLE MODE
060516 012700 060674            MOV     #XPAT00,R0      ;SET AC0 TO NON-ZERO
060522 172410                   LDD     (R0),AC0
060524 012700 060704            MOV     #XPAT10,R0      ;FSRC=0
060530 173010           X12:    SUBD    (R0),AC0        ;TEST INSTRUCTION
060532 170205                   STFPS   R5
060534 170011                   SETD
060536 012700 060664            MOV     #XDAT00,R0      ;GET RESULT
060542 174010                   STD     AC0,(R0)
060544 012701 060674            MOV     #XPAT00,R1
060550 012702 000004            MOV     #4,R2
060554 022021           X13:    CMP     (R0)+,(R1)+     ;IS RESULT CORRECT?
060556 001401                   BEQ     X14
060560 104000                   EMT                     ;
060562 077204           X14:    SOB     R2,X13
060564 012704 000200            MOV     #200,R4         ;IS FPS CORRECT?
060570 020405                   CMP     R4,R5
060572 001401                   BEQ     X15
060574 104000                   EMT                     ;
060576 012700 000200    X15:    MOV     #200,R0
060602 170100                   LDFPS   R0              ;SET DOUBLE MODE
060604 012700 060714            MOV     #XPAT20,R0      ;SET AC0=A NEGATIVE
060610 172410                   LDD     (R0),AC0
060612 012700 060704            MOV     #XPAT10,R0      ;FSRC=0
060616 173010           X16:    SUBD    (R0),AC0        ;TEST INSTRUCTION.
060620 170205                   STFPS   R5
060622 170011                   SETD
060624 012700 060664            MOV     #XDAT00,R0      ;GET RESULT
060630 174010                   STD     AC0,(R0)
060632 012701 060714            MOV     #XPAT20,R1
060636 012702 000004            MOV     #4,R2
060642 022021           X17:    CMP     (R0)+,(R1)+     ;IS RESULT CORRECT?
060644 001401                   BEQ     X20
060646 104000                   EMT                     ;
060650 077204           X20:    SOB     R2,X17
060652 012704 000210            MOV     #210,R4         ;IS FPS CORRECT?
060656 020405                   CMP     R4,R5
060660 00142°                   BEQ     XDONE
060662 104000                   EMT                     ;
060664 000000           XDAT00: .WORD   0
060666 000000           XDAT01:         0
060670 000000           XDAT02:         0
060672 000000           XDAT03:         0

060674 010421           XPAT00: .WORD   010421
060676 021042           XPAT01:         021042
060700 031463           XPAT02:         031463
060702 042104           XPAT03:         042104

060704 000000           XPAT10: .WORD   0
060706 000000           XAPT11:         0
```

```
060710  000000                          XPAT12:         0
060712  000000                          XPAT13:         0
060714  104210                          XPAT20: .WORD   104210
060716  114631                          XPAT21:         114631
060720  125252                          XPAT22:         125252
060722  135673                          XPAT23:         135673

060724                                  XDONE:
060724  004767  037544                          JSR     PC,.RSET        ;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?).

                                        ;*******************************************************************************
                                        ;TEST 446       SUBD WITH AC=0 TEST
                                        ;*******************************************************************************
060730                                  TS446:
060730  005037  061104                          CLR     @#YFLAG
060734  012737  061124  061106                  MOV     #YPAT00,@#YTMP1 ;P
060742  012737  061134  061110                  MOV     #YPAT10,@#YTMP2 ;N
060750  012737  000210  061112                  MOV     #210,@#YTMP3
060756  012700  000200                  Y1:     MOV     #200,R0
060762  170100                                  LDFPS   R0              ;SET DOUBLE MODE
060764  012700  061144                          MOV     #YPAT20,R0      ;SET AC0=0
060770  172410                                  LDD     (R0),AC0
060772  013700  061106                          MOV     @#YTMP1,R0
060776  173010                          Y2:     SUBD    (R0),AC0        ;TEST INSTRUCTION
061000  170205                                  STFPS   R5
061002  170011                                  SETD
061004  012700  061114                          MOV     #YDAT00,R0      ;GET RESULT
061010  174010                                  STD     AC0,(R0)
061012  012702  000004                          MOV     #4,R2
061016  013701  061110                          MOV     @#YTMP2,R1      ;CHECK RESULT.
061022  022021                          Y3:     CMP     (R0)+,(R1)+
061024  001401                                  BEQ     1$
061026  104000                                  EMT                     ;
061030  077204                          1$:     SOB     R2,Y3
061032  023705  061112                          CMP     @#YTMP3,R5      ;FPS CORRECT?
061036  001401                                  BEQ     Y4
061040  104000                                  EMT                     ;
061042  005737  061104                  Y4:     TST     @#YFLAG         ;FINISHED TEST?
061046  001015                                  BNE     Y5
061050  012737  177777  061104                  MOV     #-1,@#YFLAG
061056  012737  061134  061106                  MOV     #YPAT10,@#YTMP1
061064  012737  061124  061110                  MOV     #YPAT00,@#YTMP2
061072  012737  000200  061112                  MOV     #200,@#YTMP3
061100  000726                                  BR      Y1
061102  000424                          Y5:     BR      YDONE

061104  000000                          YFLAG:  .WORD   0
061106  000000                          YTMP1:          0
061110  000000                          YTMP2:          0
061112  000000                          YTMP3:          0

061114  000000                          YDAT00: .WORD   0
```

```
061116  000000                          YDAT01:         0
061120  000000                          YDAT02:         0
061122  000000                          YDAT03:         0

061124  063146                          YPAT00:         063146
061126  052525                          YPAT01:         052525
061130  042104                          YPAT02:         042104
061132  167356                          YPAT03:         167356

061134  163146                          YPAT10:         163146
061136  052525                          YPAT11:         052525
061140  042104                          YPAT12:         042104
061142  167356                          YPAT13:         167356

061144  000000                          YPAT20:         0
061146  000000                          YPAT21:         0
061150  000000                          YPAT22:         0
061152  000000                          YPAT23:         0

061154                                  YDONE:
061154  004767  037314                          JSR     PC,.RSET        ;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?).
                                        ;*************************************************************************
                                        ;TEST 447       ADDD WITH AC=0 TEST
                                        ;*************************************************************************
061160                                  TS447:
061160  005067  000134                          CLR     ZFLAG
061164  012737  061336  061322                  MOV     #ZPAT00,@#ZTMP1 ;P
061172  012737  000200  061324                  MOV     #200,@#ZTMP2
061200  012700  000200                  Z1:     MOV     #200,R0
061204  170100                                  LDFPS   R0              ;SET DOUBLE MODE
061206  012700  061356                          MOV     #ZPAT20,R0      ;SET AC0=0
061212  172410                                  LDD     (R0),AC0
061214  013700  061322                          MOV     @#ZTMP1,R0
061220  172010                          Z2:     ADDD    (R0),AC0        ;TEST INSTRUCTION
061222  170205                                  STFPS   R5
061224  170011                                  SETD
061226  012700  061326                          MOV     #ZDAT00,R0      ;GET RESULT
061232  174010                                  STD     AC0,(R0)
061234  012702  000004                          MOV     #4,R2
061240  013701  061322                          MOV     @#ZTMP1,R1      ;RESULT CORRECT?
061244  022021                          Z3:     CMP     (R0)+,(R1)+
061246  001401                                  BEQ     Z4
061250  104000                                  EMT                     ;
061252  077204                          Z4:     SOB     R2,Z3
061254  023705  061324                          CMP     @#ZTMP2,R5      ;FPS CORRECT?
061260  001401                                  BEQ     Z5
061262  104000                                  EMT                     ;
061264  005737  061320                  Z5:     TST     @#ZFLAG         ;FINISHED TEST?
061270  001012                                  BNE     Z6
061272  012737  177777  061320                  MOV     #-1,@#ZFLAG
061300  012737  061346  061322                  MOV     #ZPAT10,@#ZTMP1
061306  012737  000210  061324                  MOV     #210,@#ZTMP2
```

```
061314  000731                          BR      Z1
061316  000423                  Z6:     BR      ZDONE

061320  000000                  ZFLAG:  .WORD   0
061322  000000                  ZTMP1:          0
061324  000000                  ZTMP2:          0

061326  000000                  ZDAT00: .WORD   0
061330  000000                  ZDAT01:         0
061332  000000                  ZDAT02:         0
061334  000000                  ZDAT03:         0

061336  031463                  ZPAT00:         031463
061340  010421                  ZPAT01:         010421
061342  146314                  ZPAT02:         146314
061344  156735                  ZPAT03:         156735

061346  156735                  ZPAT10:         156735
061350  167356                  ZPAT11:         167356
061352  135673                  ZPAT12:         135673
061354  146314                  ZPAT13:         146314

061356  000000                  ZPAT20:         0
061360  000000                  ZPAT21:         0
061362  000000                  ZPAT22:         0
061364  000000                  ZPAT23:         0

061366                          ZDONE:
061366  004767  037102                  JSR     PC,.RSET        ;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?).



                                ;*******************************************************************************
                                ;TEST 450       ADDF AND ADDD WITH E(AC)=E(FSRC) TEST AND (BUT FT) TEST
                                ;*******************************************************************************
061372                          TS450:
061372  012700  003240                  MOV     #3240,R0
061376  170100                          LDFPS   R0              ;SET FIU FIV FD AND FT
061400  012700  061602                  MOV     #AAPAT0,R0      ;FLOWS IN TRAP WILL
                                                                ;OCCUR
061404  172410                          LDD     (R0),AC0        ;SET UP AC0
061406  012700  061612                  MOV     #AAPAT1,R0
061412  172010                  AA2:    ADDD    (R0),AC0        ;TEST INSTRUCTION
                                                                ;SHOULD TRUNCATE

061414  012700  061572          AA3:    MOV     #AADAT0,R0
061420  174010                          STD     AC0,(R0)        ;GET THE RESULT
061422  012701  061622                  MOV     #AAPAT2,R1
061426  012702  000004                  MOV     #4,R2
061432  022021                  AA4:    CMP     (R0)+,(R1)+     ;CORRECT?
061434  001401                          BEQ     AA7
061436  104000                          EMT                     ;
061440  077204                  AA7:    SOB     R2,AA4
```

```
                                ;NOW TEST DOUBLE FLOATING ROUND MODE.
                                ;A 1 SHOULD BE ADDED TO THE LSB ON ROUNG MODE.

061442  012700  003200                  MOV     #3200,R0        ;SET FD FIV FIV. FT=0
061446  170100                          LDFPS   R0
061450  012700  061602                  MOV     #AAPAT0,R0
061454  172410                          LDD     (R0),AC0        ;SET UP AC0 OPERAND
061456  012700  061612                  MOV     #AAPAT1,R0
061462  172010                  AA11:   ADDD    (R0),AC0        ;TEST INSTRUCTION
                                                                ;SHOULD ROUND
061464  012700  061572          AA12:   MOV     #AADAT0,R0
061470  174010                          STD     AC0,(R0)        ;GET THE RESULT
061472  012701  061632                  MOV     #AAPAT3,R1
061476  012702  000004                  MOV     #4,R2
061502  022021                  AA13:   CMP     (R0)+,(R1)+     ;CORRECT?
061504  001401                          BEQ     AA20
061506  104000                          EMT                     ;
061510  077204                  AA20:   SOB     R2,AA13

                                ;NOW TEST ADDF WITH FT=0, ROUND MODE

061512  012700  003200                  MOV     #3200,R0                ;FIV=1, FIV=1, FT=0
061516  170100                          LDFPS   R0
061520  012700  061602                  MOV     #AAPAT0,R0      ;LOAD AC0 OPERAND
061524  172410                          LDD     (R0),AC0
061526  170001                          SETF                    ;ENTER FLOATING MODE
061530  012700  061652                  MOV     #AAPAT5,R0
061534  172010                  AA22:   ADDF    (R0),AC0        ;TEST INSTRUCTION
                                                                ;SHOULD ROUND
061536                          AA23:
061536  170011                          SETD                    ;RESET TO DOUBLE
                                                                ;MODE
061540  012700  061572                  MOV     #AADAT0,R0      ;GET THE RESULT
061544  174010                          STD     AC0,(R0)
061546  012701  061662                  MOV     #AAPAT6,R1      ;CORRECT?
061552  012702  000002                  MOV     #2,R2
061556  022021                  AA24:   CMP     (R0)+,(R1)+
061560  001401                          BEQ     AA27
061562  104000                          EMT                     ;
061564  077204                  AA27:   SOB     R2,AA24
061566  000137  061672                  JMP     @#AADONE
061572  000000                  AADAT0:         0
061574  000000                                  0
061576  000000                                  0
061600  000000                                  0
061602  000200                  AAPAT0:         200
061604  000000                                  0
061606  000000                                  0
061610  000000                                  0
061612  000200                  AAPAT1:         200
061614  000000                                  0
061616  000000                                  0
061620  000001                                  1
061622  000400                  AAPAT2:         400
061624  000000                                  0
061626  000000                                  0
```

```
061630  000000                                  0
061632  000400                  AAPAT3:         400
061634  000000                                  0
061636  000000                                  0
061640  000001                                  1
061642  000400                  AAPAT4:         400
061644  000000                                  0
061646  100000                                  100000
061650  000000                                  0
061652  000200                  AAPAT5:         200
061654  000001                                  1
061656  000000                                  0
061660  000000                                  0
061662  000400                  AAPAT6:         400
061664  000001                                  1
061666  000000                                  0
061670  000000                                  0
061672                          AADONE:
061672  004767  036576                  JSR     PC,.RSET        ;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?).
                        ;*************************************************************************
                        ;TEST 451       ADDF AND ADDD WITH E(AC) LESS THAN E(FSRC) TEST
                        ;*************************************************************************
061676                  TS451:
                        ;EXPONENT DIFFERENCE=57=71 (OCT) FD=1
061676  012704  003200                  MOV     #3200,R4        ;SET FIV,FIV, AND FD
061702  170104                          LDFPS   R4
061704  012700  062352                  MOV     #CCP0,R0        ;SET AC0 OPERAND
061710  172410                          LDD     (R0),AC0                ;AC0
061712  012700  062372                  MOV     #CCP2,R0
061716  172010                  CCX2:   ADDD    (R0),AC0        ;TEST INSTRUCTION
061720  170205                          STFPS   R5              ;GET FPS
061722  012700  062342                  MOV     #CCDAT0,R0      ;GET THE RESULT
061726  174010                          STD     AC0,(R0)
061730  012701  062372                  MOV     #CCP2,R1        ;IS IT CORRECT
061734  012702  000004                  MOV     #4,R2
061740  022021                  CCX3:   CMP     (R0)+,(R1)+
061742  001401                          BEQ     CCX6
061744  104000                          EMT                     ;
061746  077204                  CCX6:   SOB     R2,CCX3
061750  020405                          CMP     R4,R5           ;FPS CORRECT?
061752  001401                          BEQ     CCX7
061754  104000                          EMT                     ;
                        ;EXPONENT DIFFERENCE=56=70 (OCT) FD=1
061756  012704  003200          CCX7:   MOV     #3200,R4        ;SET FIV,FIV, AND FD
061762  170104                          LDFPS   R4
061764  012700  062352                  MOV     #CCP0,R0        ;SET AC0 OPERAND
061770  172410                          LDD     (R0),AC0
061772  012700  062362                  MOV     #CCP1,R0        ;FSRC
061776  172010                  CCX8:   ADDD    (R0),AC0        ;TEST INSTRUCTION
062000  170205                          STFPS   R5              ;GET FPS
062002  012700  062342                  MOV     #CCDAT0,R0      ;GET THE RESULT
062006  174010                          STD     AC0,(R0)
```

```
062010 012701 062442            MOV     #CCP7,R1        ;IS IT CORRECT
062014 012702 000004            MOV     #4,R2
062020 022021           CCX9:   CMP     (R0)+,(R1)+
062022 001401                   BEQ     CCX12
062024 104000                   EMT                     ;
062026 077204           CCX12:  SOB     R2,CCX9
062030 020405                   CMP     R4,R5           ;FPS CORRECT?
062032 001401                   BEQ     CCX13
062034 104000                   EMT                     ;
                        ;EXPONENT DIFFERENCE=25=31 (OCT) FD=0
062036 012700 062352    CCX13:  MOV     #CCP0,R0        ;SET UP AC0 OPERAND.
062042 172410                   LDD     (R0),AC0
062044 012704 003000            MOV     #3000,R4        ;SET FIV,FIV. CLEAR FD.
062050 170104                   LDFPS   R4
062052 012700 062432            MOV     #CCP6,R0        ;FSRC
062056 172010           CCX14:  ADDF    (R0),AC0        ;TEST INSTRUCTION
062060 170205                   STFPS   R5
062062 170011                   SETD                    ;REENTER DOUBLE MOVE
062064 012700 062342            MOV     #CCDAT0,R0      ;GET THE RESULT
062070 174010                   STD     AC0,(R0)
062072 012701 062432            MOV     #CCP6,R1        ;IS THE RESULT CORRECT?
062076 012702 000002            MOV     #2,R2
062102 022021           CCX15:  CMP     (R0)+,(R1)+
062104 001401                   BEQ     CCX18
062106 104000                   EMT                     ;
062110 077204           CCX18:  SOB     R2,CCX15
062112 020405                   CMP     R4,R5
062114 001401                   BEQ     CCX19
062116 104000                   EMT                     ;
                        ;EXPONENT DIFFERENCE=24=30 (OCT) FD=0
062120 012700 062402    CCX19:  MOV     #CCP3,R0        ;SET UP AC0 OPERAND.
062124 172410                   LDD     (R0),AC0
062126 012704 003000            MOV     #3000,R4        ;SET FIV,FIV. CLEAR FD.
062132 170104                   LDFPS   R4
062134 012700 062422            MOV     #CCP5,R0        ;FSRC
062140 172010           CCX20:  ADDF    (R0),AC0        ;TEST INSTRUCTION
062142 170205                   STFPS   R5
062144 170011                   SETD                    ;REENTER DOUBLE MOVE
062146 012700 062342            MOV     #CCDAT0,R0      ;GET THE RESULT
062152 174010                   STD     AC0,(R0)
062154 012701 062452            MOV     #CCP10,R1       ;IS THE RESLT CORRECT?
062160 012702 000002            MOV     #2,R2
062164 022021           CCX21:  CMP     (R0)+,(R1)+
062166 001401                   BEQ     CCX24
062170 104000                   EMT                     ;
062172 077204           CCX24:  SOB     R2,CCX21
062174 020405                   CMP     R4,R5
062176 001401                   BEQ     CCX25
062200 104000                   EMT                     ;
                        ;EXPONENT DIFFERENCE=1 FD=1
062202 012704 003200    CCX25:  MOV     #3200,R4        ;SET FIV,FIV, AND FD
062206 170104                   LDFPS   R4
062210 012700 062352            MOV     #CCP0,R0        ;SET AC0 OPERAND
062214 172410                   LDD     (R0),AC0
062216 012700 062402            MOV     #CCP3,R0        ;FSRC
062222 172010           CCX26:  ADDD    (R0),AC0        ;TEST INSTRUCTION
```

```
062224  170205                          STFPS   R5              ;GET FPS
062226  012700  062342                  MOV     #CCDAT0,R0      ;GET THE RESULT
062232  174010                          STD     AC0,(R0)
062234  012701  062462                  MOV     #CCP11,R1       ;IS IT CORRECT
062240  012702  000004                  MOV     #4,R2
062244  022021                  CCX27:  CMP     (R0)+,(R1)+
062246  001401                          BEQ     CCX30
062250  104000                          EMT                     ;
062252  077204                  CCX30:  SOB     R2,CCX27
062254  020405                          CMP     R4,R5           ;FPS CORRECT?
062256  001401                          BEQ     CCX31
062260  104000                          EMT                     ;
                                ;EXPONENT DIFFERENCE=100=144 (OCT) FD=1
062262  012704  003200          CCX31:  MOV     #3200,R4        ;SET FIV,FIV, AND FD
062266  170104                          LDFPS   R4
062270  012700  062352                  MOV     #CCP0,R0        ;SET AC0 OPERAND
062274  172410                          LDD     (R0),AC0
062276  012700  062412                  MOV     #CCP4,R0        ;FSRC
062302  172010                  CCX32:  ADDD    (R0),AC0        ;TEST INSTRUCTION
062304  170205                          STFPS   R5              ;GET FPS
062306  012700  062342                  MOV     #CCDAT0,R0      ;GET THE RESULT
062312  174010                          STD     AC0,(R0)
062314  012701  062412                  MOV     #CCP4,R1        ;IS IT CORRECT
062320  012702  000004                  MOV     #4,R2
062324  022021                  CCX33:  CMP     (R0)+,(R1)+
062326  001401                          BEQ     CCX36
062330  104000                          EMT                     ;
062332  077204                  CCX36:  SOB     R2,CCX33
062334  020405                          CMP     R4,R5           ;FPS CORRECT?
062336  001461                          BEQ     CCXDONE
062340  104000                          EMT                     ;
062342  000000                  CCDAT0: 0
062344  000000                          0
062346  000000                          0
062350  000000                          0
062352  000200                  CCP0:   200                     ;E(AC)=1
062354  000000                          0
062356  000000                          0
062360  000000                          0
062362  016200                  CCP1:   16200                   ;E(FSRC)=E(AC)+56=57
062364  000000                          0                       ;                =71(OCT)
062366  000000                          0
062370  000000                          0
062372  016400                  CCP2:   16400                   ;E(FSRC)=E(AC)+57=58
062374  000000                          0                       ;                =72(OCT)
062376  000000                          0
062400  000000                          0
062402  000400                  CCP3:   400                     ;E(FSRC)=E(AC)+1=2
062404  000000                          0
062406  000000                          0
062410  000000                          0
062412  031200                  CCP4:   31200                   ;E(FSRC)=E(AC)+100=101=145(OCT)
062414  000000                          0
062416  000000                          0
062420  000000                          0
062422  006200                  CCP5:   6200                    ;E(FSRC)=E(AC)+24=25=31(OCT)
```

```
062424  000000                        0
062426  000000                        0
062430  000000                        0
062432  006400                CCP6:   6400                          ;E(FSRC)=E(AC)+25=26=32(OCT)
062434  000000                        0
062436  000000                        0
062440  000000                        0
062442  016200                CCP7:   16200                         ;CCP1 RES
062444  000000                        0
062446  000000                        0
062450  000001                        1
062452  006200                CCP10:  6200                          ;CCP5 RES
062454  000001                        1
062456  000000                        0
062460  000000                        0
062462  000500                CCP11:  500                           ;CCP3 RES
062464  000000                        0
062466  000000                        0
062470  000000                        0
062472  000200                CCP12:  200                           ;BAD CONSTANT
062474  000000                        0                             ;RES CCP2,CCP4
062476  000000                        0
062500  000000                        0

062502                        CCXDONE:
062502  004767  035766                JSR     PC,.RSET              ;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?).

                              ;*************************************************************************
                              ;TEST 452       ADDF AND ADDD WITH E(AC) GREATER THAN E(FSRC) TEST
                              ;*************************************************************************
062506                        TS452:
                              ;EXPONENT DIFFERENCE=57=71 (OCT) FD=1
062506  012704  003200                MOV     #3200,R4              ;SET FIV FIV, AND FD
062512  170104                        LDFPS   R4
062514  012700  063202                MOV     #BBPAT2,R0            ;SET AC0 OPERAND.
062520  172410                        LDD     (R0),AC0
062522  012700  063172                MOV     #BBPAT1,R0            ;FSRC
062526  172010                BB2:    ADDD    (R0),AC0              ;TEST INSTRUCTION
062530  170205                        STFPS   R5
062532  012700  063152        BB3:    MOV     #BBDAT0,R0            ;GET THE RESULT
062536  174010                        STD     AC0,(R0)
062540  012701  063202                MOV     #BBPAT2,R1            ;RESULT CORRECT?
062544  012702  000004                MOV     #4,R2
062550  022021                BB4:    CMP     (R0)+,(R1)+
062552  001401                        BEQ     BB5
062554  104000                        EMT                           ;
062556  077204                BB5:    SOB     R2,BB4
                                                                    ;WAS FPS CORRECT?
062560  020405                        CMP     R4,R5
062562  001401                        BEQ     BB6
062564  104000                        EMT                           ;
                              ;EXPONENT DIFFERENCE=56=70 (OCT) FD=1
```

```
062566  012704  003200    BB6:    MOV     #3200,R4        ;SET FIV,FIV, AND FD
062572  170104                    LDFPS   R4
062574  012700  063222            MOV     #BBPAT4,R0      ;SET AC0 OPERAND
062600  172410                    LDD     (R0),AC0
062602  012700  063172            MOV     #BBPAT1,R0      ;FSRC
062606  172010            BB7:    ADDD    (R0),AC0        ;TEST INSTRUCTION
062610  170205                    STFPS   R5              ;GET FPS
062612  012700  063152            MOV     #BBDAT0,R0      ;GET THE RESULT
062616  174010                    STD     AC0,(R0)
062620  012701  063262            MOV     #BBP10,R1       ;IS IT CORRECT
062624  012702  000004            MOV     #4,R2
062630  022021            BB10:   CMP     (R0)+,(R1)+
062632  001401                    BEQ     BB13
062634  104000                    EMT                     ;
062636  077204            BB13:   SOB     R2,BB10
062640  020405                    CMP     R4,R5           ;FPS CORRECT?
062642  001401                    BEQ     BB14
062644  104000                    EMT                     ;
                          ;EXPONENT DIFFERENCE=25=31 (OCT) FD=0
062646  012700  063162    BB14:   MOV     #BBPAT0,R0      ;SET UP AC0 OPERAND
062652  172410                    LDD     (R0),AC0
062654  012704  003000            MOV     #3000,R4        ;SET FIV AND FIV
                                                          ;CLEAR FD
062660  170104                    LDFPS   R4
062662  012700  063172            MOV     #BBPAT1,R0      ;FSRC
062666  172010            BB15:   ADDF    (R0),AC0        ;TEST INSTRUCTION
062670  170205                    STFPS   R5
062672  170011                    SETD                    ;REENTED DOUBLE MODE.
062674  012700  063152            MOV     #BBDAT0,R0      ;GET THE RESULT
062700  174010                    STD     AC0,(R0)
062702  012701  063162            MOV     #BBPAT0,R1      ;IS THE RESULT
062706  012702  000002            MOV     #2,R2           ;CORRECT?
062712  022021            BB16:   CMP     (R0)+,(R1)+
062714  001401                    BEQ     BB17
062716  104000                    EMT                     ;
062720  077204            BB17:   SOB     R2,BB16
062722  020405                    CMP     R4,R5           ;IS FPS CORRECT?
062724  001401                    BEQ     BB20
062726  104000                    EMT                     ;
                          ;EXPONENT DIFFERENCE=24=30 (OCT)
062730  012700  063212    BB20:   MOV     #BBPAT3,R0      ;SET UP AC0 OPERAND.
062734  172410                    LDD     (R0),AC0
062736  012704  003000            MOV     #3000,R4        ;SET FIU,FIV. CLEAR FD.
062742  170104                    LDFPS   R4
062744  012700  063172            MOV     #BBPAT1,R0      ;FSRC
062750  172010            BB21:   ADDF    (R0),AC0        ;TEST INSTRUCTION
062752  170205                    STFPS   R5
062754  170011                    SETD                    ;REENTER DOUBLE MODE
062756  012700  063152            MOV     #BBDAT0,R0      ;GET THE RESULT
062762  174010                    STD     AC0,(R0)
062764  012701  063252            MOV     #BBP7,R1        ;IS THE RESULT CORRECT?
062770  012702  000002            MOV     #2,R2
062774  022021            BB22:   CMP     (R0)+,(R1)+
062776  001401                    BEQ     BB25
063000  104000                    EMT                     ;
063002  077204            BB25:   SOB     R2,BB22
```

```
063004  020405                          CMP     R4,R5
063006  001401                          BEQ     BB26
063010  104000                          EMT                     ;
                                ;EXPONENT DIFFERENCE=1
063012  012704  003200          BB26:   MOV     #3200,R4
063016  170104                          LDFPS   R4              ;SET UP AC0 OPERAND
063020  012700  063232                  MOV     #BBPAT5,R0
063024  172410                          LDD     (R0),AC0
063026  012700  063172                  MOV     #BBPAT1,R0      ;FSRC
063032  172010                  BB27:   ADDD    (R0),AC0        ;TEST INSTRUCTION
063034  170205                          STFPS   R5
063036  012700  063152                  MOV     #BBDAT0,R0      ;GET THE RESULT.
063042  174010                          STD     AC0,(R0)
063044  012701  063272                  MOV     #BBP11,R1       ;IS IT CORRECT?
063050  012702  000004                  MOV     #4,R2
063054  022021                  BB30:   CMP     (R0)+,(R1)+
063056  001401                          BEQ     BB31
063060  104000                          EMT                     ;
063062  077204                  BB31:   SOB     R2,BB30
063064  020405                          CMP     R4,R5           ;IS FPS CORRECT
063066  001401                          BEQ     BB32
063070  104000                          EMT                     ;
                                ;EXPONENT DIFFERENCE=100=144 (OCT)
063072  012704  003200          BB32:   MOV     #3200,R4
063076  170104                          LDFPS   R4              ;SET FIV,FIV AND FD
063100  012700  063242                  MOV     #BBPAT6,R0      ;SET UP AC0 OPERAND.
063104  172410                          LDD     (R0),AC0
063106  012700  063172                  MOV     #BBPAT1,R0      ;FSRC
063112  172010                  BB33:   ADDD    (R0),AC0        ;TEST INSTRUCTION
063114  170205                          STFPS   R5
063116  012700  063152                  MOV     #BBDAT0,R0      ;GET THE RESULT
063122  174010                          STD     AC0,(R0)
063124  012701  063242                  MOV     #BBPAT6,R1      ;IS IT CORRECT
063130  012702  000004                  MOV     #4,R2
063134  022021                  BB34:   CMP     (R0)+,(R1)+
063136  001401                          BEQ     BB35
063140  104000                          EMT                     ;
063142  077204                  BB35:   SOB     R2,BB34
063144  020405                          CMP     R4,R5           ;IS FPS CORRECT
063146  001455                          BEQ     BBDONE
063150  104000                          EMT                     ;
063152  000000                  BBDAT0: 0
063154  000000                          0
063156  000000                          0
063160  000000                          0
063162  006400                  BBPAT0: 6400                    ;F(AC)=E(FSRC)+25=26
063164  000000                          0                       ;               =32(OCT)
063166  000000                          0
063170  000000                          0
063172  000200                  BBPAT1: 200
063174  000000                          0                       ;E(FSRC)=1
063176  000000                          0
063200  000000                          0
063202  016400                  BBPAT2: 16400
063204  000000                          0                       ;E(AC)=E(FSRC)+57=58
063206  000000                          0                       ;               =72(OCT)
```

```
063210  000000                  0
063212  006200          BBPAT3: 6200                    ;E(AC)=E(FSRC)+24=25
063214  000000                  0                       ;                 =31(OCT)
063216  000000                  0
063220  000000                  0
063222  016200          BBPAT4: 16200                   ;E(AC)=E(FSR6)+56=57
063224  000000                  0                       ;                 =71(OCT)
063226  000000                  0
063230  000000                  0
063232  000400          BBPAT5: 400                     ;E(AC)=E(FSRC)+1=2
063234  000000                  0
063236  000000                  0
063240  000000                  0
063242  031200          BBPAT6: 31200                   ;E(AC)=E(FSRC)+100=101
063244  000000                  0                       ;                  =145(OCT)
063246  000000                  0
063250  000000                  0
063252  006200          BBP7:   6200                    ;BBPAT3 RES
063254  000001                  1
063256  000000                  0
063260  000000                  0
063262  016200          BBP10:  16200                   ;BBPAT4 RES
063264  000000                  0
063266  000000                  0
063270  000001                  1
063272  000500          BBP11:  500                     ;BBPAT5 RES
063274  000000                  0
063276  000000                  0
063300  000000                  0
063302                  BBDONE:
063302  004767 035166           JSR     PC,.RSET        ;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?).
                        ;*******************************************************************
                        ;TEST 453       ADDD WITH NEGATVE OPRANDS TEST
                        ;*******************************************************************

063306                  TS453:
                        ;BOTH OPERANDS NEGATIVE
063306  012704 003200           MOV     #3200,R4        ;SET FIO, FIV, AND FD
063312  170104                  LDFPS   R4
063314  012700 064072           MOV     #DDP1,R0                ;SET AC0 OPERAND
063320  172410                  LDD     (R0),AC0
063322  012700 064072           MOV     #DDP1,R0                ;ESRC
063326  172010          DD2:    ADDD    (R0),AC0                ;TEST INSTRUCTION
063330  170205                  STFPS   R5                      ;GET FPS
063332  012700 064052           MOV     #DDDAT0,R0              ;GET THE RESULT
063336  174010                  STD     AC0,(R0)
063340  012701 064172           MOV     #DDP9,R1                ;IS IT CORRECT
063344  012702 000004           MOV     #4,R2
063350  022021          DD3:    CMP     (R0)+,(R1)+
063352  001401                  BEQ     DD6
063354  104000                  EMT                     ;
063356  077204          DD6:    SOB     R2,DD3
063360  052704 000010           BIS     #10,R4
```

```
063364 020405                    CMP     R4,R5              ;FPS CORRECT?
063366 001401                    BEQ     DD7
063370 104000                    EMT                ;
                         ;AC POS FSRC NEG      AC=-FSRC
063372 012704 003200     DD7:    MOV     #3200,R4           ;SET FIO, FIV, AND FD
063376 170104                    LDFPS   R4
063400 012700 064102             MOV     #DDP2,R0           ;SET AC0 OPERAND
063404 172410                    LDD     (R0),AC0
063406 012700 064072             MOV     #DDP1,R0           ;FSPC
063412 172010            DD8:    ADDD    (R0),AC0           ;TEST INSTRUCTION
063414 170205                    STFPS   R5                 ;GET FPS
063416 012700 064052             MOV     #DDDAT0,R0         ;GET THE RESULT
063422 174010                    STD     AC0,(R0)
063424 012701 064062             MOV     #DDP0,R1           ;IS IT CORRECT
063430 012702 000004             MOV     #4,R2
063434 022021            DD10:   CMP     (R0)+,(R1)+
063436 001401                    BEQ     DD11
063440 104000                    EMT                ;
063442 077204            DD11:   SOB     R2,DD10
063444 052704 000004             BIS     #4,R4
063450 020405                    CMP     R4,R5              ;FPS CORRECT?
063452 001401                    BEQ     DD12
063454 104000                    EMT                ;
                         ;AC NEG FSRC POS               AC=-FSRC
063456 012704 003200     DD12:   MOV     #3200,R4           ;SET FIU, FIV, AND FD
063462 170104                    LDFPS   R4
063464 012700 064072             MOV     #DDP1,R0           ;SET AC0 OPERAND
063470 172410                    LDD     (R0),AC0
063472 012700 064102             MOV     #DDP2,R0           ;FSRC
063476 172010            DD13:   ADDD    (R0),AC0           ;TEST INSTRUCTION
063500 170205                    STFPS   R5                 ;GET FPS
063502 012700 064052             MOV     #DDDAT0,R0         ;GET THE RESULT
063506 174010                    STD     AC0,(R0)
063510 012701 064062             MOV     #DDP0,R1           ;IS IT CORRECT
063514 012702 000004             MOV     #4,R2
063520 022021            DD14:   CMP     (R0)+,(R1)+
063522 001401                    BEQ     DD15
063524 104000                    EMT                ;
063526 077204            DD15:   SOB     R2,DD14
063530 052704 000004             BIS     #4,R4
063534 020405                    CMP     R4,R5              ;EPS CORRECT?
063536 001401                    BEQ     DD16
063540 104000                    EMT                ;
                         ;AC0 POC        FSRC    NEG    /AC/ > /FSRC/
063542 012704 003200     DD16:   MOV     #3200,R4           ;SET FIV, FIV AND FD
063546 170104                    LDFPS   R4
063550 012700 064112             MOV     #DDP3,R0           ;SET AC0 OPERAND
063554 172410                    LDD     (R0),AC0
063556 012700 064142             MOV     #DDP6,R0           ;ESPC
063562 172010            DD17:   ADDD    (R0),AC0           ;TEST INSTRUCTION
063564 170205                    STFPS   R5                 ;GET FPS
063566 012700 064052             MOV     #DDDAT0,R0         ;GET THE RESULT
063572 174010                    STD     AC0,(R0)
063574 012701 064152             MOV     #DDP7,R1           ;IS IT CORRECT
063600 012702 000004             MOV     #4,R2
063604 022021            DD18:   CMP     (R0)+,(R1)+
```

```
063606  001401                          BEQ     DD21
063610  104000                          EMT                     ;
063612  077204                  DD21:   SOB     R2,DD18
063614  020405                          CMP     R4,R5                           ;EPS CORRECT?
063616  001401                          BEQ     DD22
063620  104000                          EMT                     ;
                                ;AC NEG FSRC    POS     /FSRC/ > /AC/
063622  012704  003200          DD22:   MOV     #3200,R4                        ;SET FIO,FIV, AND FD
063626  170104                          LDFPS   R4
063630  012700  064142                  MOV     #DDP6,R0                        ;SET ACO OPERAND
063634  172410                          LDD     (R0),AC0
063636  012700  064112                  MOV     #DDP3,R0                        ;FSPC
063642  172010                  DD23:   ADDD    (R0),AC0                        ;TEST INSTRUCTION
063644  170205                          STFPS   R5                              ;GET FPS
063646  012700  064052                  MOV     #DDDAT0,R0                      ;GET THE RESULT
063652  174010                          STD     AC0,(R0)
063654  012701  064152                  MOV     #DDP7,R1                        ;IS IT CORRECT?
063660  012702  000004                  MOV     #4,R2
063664  022021                  DD24:   CMP     (R0)+,(R1)+
063666  001401                          BEQ     DD27
063670  104000                          EMT                     ;
063672  077204                  DD27:   SOB     R2,DD24
063674  020405                          CMP     R4,R5                           ;FPS CORRECT?
063676  001401                          BEQ     DD30
063700  104000                          EMT                     ;
                                ;AC0 POS        FSRC    NEG     /AC/</FRSRC/
063702  012704  003200          DD30:   MOV     #3200,R4                        ;SET FIO,FIV,AND FD
063706  170104                          LDFPS   R4
063710  012700  064122                  MOV     #DDP4,R0                        ;SET ACO OPERAND
063714  172410                          LDD     (R0),AC0
063716  012700  064132                  MOV     #DDP5,R0                        ;FSPC
063722  172010                  DD31:   ADDD    (R0),AC0                        ;TEST INSTRUCTION
063724  170205                          STFPS   R5                              ;GET FPS
063726  012700  064052                  MOV     #DDDAT0,R0                      ;GET THE RESULT
063732  174010                          STD     AC0,(R0)
063734  012701  064162                  MOV     #DDP8,R1                        ;IS IT CORRECT
063740  012702  000004                  MOV     #4,R2
063744  022021                  DD32:   CMP     (R0)+,(R1)+
063746  001401                          BEQ     DD35
063750  104000                          EMT                     ;
063752  077204                  DD35:   SOB     R2,DD32
063754  052704  000010                  BIS     #10,R4
063760  020405                          CMP     R4,R5                           ;FPS CORRECT?
063762  001401                          BEQ     DD36
063764  104000                          EMT                     ;
                                ;AC0 NEG        FSRC    POS     /FSRC/</AC/
063766  012704  003200          DD36:   MOV     #3200,R4                        ;SET FIO, FIV, AND FD
063772  170104                          LDFPS   R4
063774  012700  064132                  MOV     #DDP5,R0                        ;SET ACO OPERAND
064000  172410                          LDD     (R0),AC0
064002  012700  064122                  MOV     #DDP4,R0                        ;FSPC
064006  172010                  DD37:   ADDD    (R0),AC0                        ;TEST INSTRUCTION
064010  170205                          STFPS   R5                              ;GET FPS
064012  012700  064052                  MOV     #DDDAT0,R0                      ;GET THE RESULT
064016  174010                          STD     AC0,(R0)
064020  012701  064162                  MOV     #DDP8,R1                        ;IS IT CORRECT
```

```
064024  012702  000004          MOV     #4,R2
064030  022021          DD38:   CMP     (R0)+,(R1)+
064032  001401                  BEQ     DD41
064034  104000                  EMT                     ;
064036  077204          DD41:   SOB     R2,DD38
064040  052704  000010          BIS     #10,R4
064044  020405                  CMP     R4,R5                   ;FPS CORRECT?
064046  001455                  BEQ     DDDONE
064050  104000                  EMT                     ;
064052  000000          DDDATO: 0
064054  000000                  0
064056  000000                  0
064060  000000                  0
064062  000000          DDP0:   0
064064  000000                  0
064066  000000                  0
064070  000000                  0
064072  100200          DDP1:   100200                          ;-DDP2
064074  000000                  0
064076  000000                  0
064100  000000                  0
064102  000200          DDP2:   200                             ;-DDP1
064104  000000                  0
064106  000000                  0
064110  000000                  0
064112  001100          DDP3:   1100                            ;EXP=4
064114  000000                  0                               ;FRAC=...110...
064116  000000                  0
064120  000000                  0
064122  000600          DDP4:   600                             ;EXP=3
064124  000000                  0                               ;FRAC=...100...
064126  000000                  0
064130  000000                  0
064132  101100          DDP5:   101100                          ;-DDP3
064134  000000                  0
064136  000000                  0
064140  000000                  0
064142  100600          DDP6:   100600                          ;-DDP4
064144  000000                  0
064146  000000                  0
064150  000000                  0
064152  001000          DDP7:   1000                            ;DDP3+DDP6
064154  000000                  0
064156  000000                  0
064160  000000                  0
064162  101000          DDP8:   101000                          ;DDP5+DDP4
064164  000000                  0
064166  000000                  0
064170  000000                  0
064172  100400          DDP9:   100400                          ;DDP1+DDP1
064174  000000                  0
064176  000000                  0
064200  000000                  0
064202                  DDDONE:
064202  004767  034266          JSR     PC,.RSET        ;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?).
                            ;*************************************************************************
                            ;TEST 454       SUBD TEST
                            ;*************************************************************************
064206                      TS454:
                            ;       USE POSITIVE OPERANDS
064206  012704  003200              MOV     #3200,R4                ;SET FIU, FIV, AND FD
064212  170104                      LDFPS   R4
064214  012700  064400              MOV     #EEP1,R0                ;SET AC0 OPERAND
064220  172410                      LDD     (R0),AC0
064222  012700  064400              MOV     #EEP1,R0                ;FSPC
064226  173010              EE2:    SUBD    (R0),AC0                ;TEST INSTRUCTION
064230  170205                      STFPS   R5                      ;GET FPS
064232  012700  064356              MOV     #EEDATO,R0              ;GET THE RESULT
064236  174010                      STD     AC0,(R0)
064240  012701  064366              MOV     #EEP0,R1                ;IS IT CORRECT?
064244  012702  000004              MOV     #4,R2
064250  022021              EE3:    CMP     (R0)+,(R1)+
064252  001401                      BEQ     EE6
064254  104000                      EMT                     ;
064256  077204              EE6:    SOB     R2,EE3
064260  052704  000004              BIS     #4,R4
064264  020405                      CMP     R4,R5                   ;FPS CORRECT?
064266  001401                      BEQ     EE7
064270  104000                      EMT                     ;
                                    ;USE NEGATIVE OPERANDS
064272  012704  003200      EE7:    MOV     #3200,R4                ;SET FIO, FIV, AND FD
064276  170104                      LDFPS   R4
064300  012700  064420              MOV     #EEP3,R0                ;SET AC0 OPERAND
064304  172410                      LDD     (R0),AC0
064306  012700  064420              MOV     #EEP3,R0                ;FSPC
064312  173010              EE8:    SUBD    (R0),AC0                ;TEST INSTRUCTION
064314  170205                      STFPS   R5                      ;GET FPS
064316  012700  064356              MOV     #EEDATO,R0              ;GET THE RESULT
064322  174010                      STD     AC0,(R0)
064324  012701  064366              MOV     #EEP0,R1                ;IS IT CORRECT?
064330  012702  000004              MOV     #4,R2
064334  022021              EE9:    CMP     (R0)+,(R1)+
064336  001401                      BEQ     EE12
064340  104000                      EMT                     ;
064342  077204              EE12:   SOB     R2,EE9
064344  052704  000004              BIS     #4,R4
064350  020405                      CMP     R4,R5                   ;FPS CORRECT?
064352  001432                      BEQ     EEDONE
064354  104000                      EMT                     ;
064356  000000              EEDATO: 0
064360  000000                      0
064362  000000                      0
064364  000000                      0
064366  000000              EEP0:   0
064370  000000                      0
064372  000000              00000
064374  000000                      0
064376  000000                      0
```

```
064400 000200                 EEP1:   200
064402 000000                         0
064404 000000                         0
064406 000000                         0
064410 000400                 EEP2:   400
064412 000000                         0
064414 000000                         0
064416 000000                         0
064420 100200                 EEP3:   100200
064422 000000                         0
064424 000000                         0
064426 000000                         0
064430 100400                 EEP4:   100400
064432 000000                         0
064434 000000                         0
064436 000000                         0
064440                        EEDONE:
064440 004767 034030                  JSR     PC,.RSET        ;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?).
                              ;*********************************************************************
                              ;TEST 455         NORMALIZE ALGORITHM TEST
                              ;*********************************************************************
064444                        TS455:
                              ;USE DATA PATTERNS THAT REQUIRE ONLY ONE LEFT SHIFT TO NORMALIZE
064444 012704 003200                  MOV     #3200,R4                ;SET FIO, FIV, AND FD
064450 170104                         LDFPS   R4
064452 012700 064634                  MOV     #FFP2,R0                ;SET AC0 OPERAND
064456 172410                         LDD     (R0),AC0
064460 012700 064644                  MOV     #FFP3,R0                ;FSPC
064464 172010                 FF2:    ADDD    (R0),AC0                ;TEST INSTRUCTION
064466 170205                         STFPS   R5                      ;GET FPS
064470 012700 064604                  MOV     #FFDAT0,R0              ;GET THE RESULT
064474 174010                         STD     AC0,(R0)
064476 012701 064654                  MOV     #FFP4, R1               ;IS IT CORRECT
064502 012702 000004                  MOV     #4,R2
064506 022021                 FF3:    CMP     (R0)+,(R1)+
064510 001401                         BEQ     FF4
064512 104000                         EMT                     ;
064514 077204                 FF4:    SOB     R2,FF3
064516 020405                         CMP     R4,R5                   ;FPS CORRECT?
064520 001401                         BEQ     FF5
064522 104000                         EMT                     ;
                              ;USE DATA PATTERNS WHICH REQUIRE 56 LEFT SHIFTS TO NORMALIZE
                              ;THE RESULT
064524 012704 003200          FF5:    MOV     #3200,R4                ;SET FIU, FIV, AND FD
064530 170104                         LDFPS   R4
064532 012700 064614                  MOV     #FFP0,R0                ;SET AC0 OPERAND
064536 172410                         LDD     (R0),AC0
064540 012700 064624                  MOV     #FFP1,R0                ;FSRC
064544 172010                 FF6:    ADDD    (R0),AC0                ;TEST INSTRUCTION
064546 170205                         STFPS   R5                      ;GET FPS
064550 012700 064604                  MOV     #FFDAT0,R0              ;GET THE RESULT
064554 174010                         STD     AC0,(R0)
```

```
064556 012701 064654           MOV     #FFP4,R1              ;IS IT CORRECT
064562 012702 000004           MOV     #4,R2
064566 022021          FF7:    CMP     (R0)+,(R1)+
064570 001401                  BEQ     FF10
064572 104000                  EMT                   ;
064574 077204          FF10:   SOB     R2,FF7
064576 020405                  CMP     R4,R5                 ;FPS CORRECT?
064600 001431                  BEQ     FFDONE
064602 104000                  EMT                   ;


064604 000000          FFDATO: 0
064606 000000                  0
064610 000000                  0
064612 000000                  0

064614 016000          FFP0:   16000
064616 000000                  0
064620 000000                  0
064622 000001                  1
064624 116000          FFP1:   116000
064626 000000                  0
064630 000000                  0
064632 000000                  0
064634 000500          FFP2:   500
064636 000000                  0
064640 000000                  0
064642 000000                  0
064644 100400          FFP3:   100400
064646 000000                  0
064650 000000                  0
064652 000000                  0
064654 000200          FFP4:   200                           ;FFP4=FFP0+FFP1
064656 000000                  0                             ;    =FFP3+FFP4
064660 000000                  0
064662 000000                  0

064664                 FFDONE:
064664 004767 033604           JSR     PC,.RSET      ;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?).




                       ;*****************************************************************
                       ;*****************************************************************
                       ;FLOATING POINT SECOND PART
                       ;*****************************************************************
                       ;*****************************************************************


                       ;*******************************************************************************
                       ;TEST 456      ROUND\TRUNK TEST
                       ;*******************************************************************************
```

```
064670                          TS456:

                                        ;ROUND AND NORMALIZE TEST

064670  012704  003200                  MOV     #3200,R4                ;SET FIU, FIV, AND FD
064674  170104                          LDFPS   R4
064676  012700  065300                  MOV     #HHP0,R0                ;SET AC0 OPERAND
064702  172410                          LDD     (R0),AC0
064704  012700  065310                  MOV     #HHP1,R0                ;FSPC
064710  172010                  HH2:    ADDD    (R0),AC0                ;TEST INSTRUCTION
064712  170205                          STFPS   R5                      ;GET FPS
064714  012700  065270                  MOV     #HHDAT0,R0              ;GET THE RESULT
064720  174010                          STD     AC0,(R0)
064722  012701  065320                  MOV     #HHP2,R1                ;IS IT CORRECT
064726  012702  000004                  MOV     #4,R2
064732  022021                  HH3:    CMP     (R0)+,(R1)+
064734  001401                          BEQ     HH6
064736  104000                          EMT                     ;
064740  077204                  HH6:    SOB     R2,HH3
064742  020405                          CMP     R4,R5           ;FPS CORRECT?
064744  001401                          BEQ     HH7
064746  104000                          EMT                     ;

                                ;THIS IS A TEST OF THE ABILITY
                                ;OF NORMALIZE TO PRODUCE A ZERO EXP. AND
                                ;OF THE R\T ALGORITHM TO PROPERLY SET THE FPS

064750  012704  043200          HH7:    MOV     #043200,R4              ;SET FIU,FIV,AND FD
                                                                        ;FID
064754  170104                                  LDFPS   R4
064756  012700  065350                  MOV     #HHP5,R0                ;SET AC0 OPERAND
064762  172410                          LDD     (R0),AC0
064764  012700  065360                  MOV     #HHP6,R0                ;FSPC
064770  172010                  HH8:    ADDD    (R0),AC0                ;TEST INS RUCTION
064772  170205                          STFPS   R5                      ;GET FPS
064774  012700  065270                  MOV     #HHDAT0,R0              ;GET THE RESULT
065000  174010                          STD     AC0,(R0)
065002  012701  065340                  MOV     #HHP4,R1                ;IS IT CORRECT
065006  012702  000004                  MOV     #4,R2
065012  022021                  HH9:    CMP     (R0)+,(R1)+
065014  001401                          BEQ     HH10
065016  104000                          EMT                     ;
065020  077204                  HH10:   SOB     R2,HH9
065022  052704  100004                  BIS     #100004,R4
065026  020405                          CMP     R4,R5
065030  001401                          BEQ     HH11
065032  104000                          EMT                     ;

                                ;THIS IS A TEST OF THE R\T ALGORITHN'S
                                ;ABILITY TO SET BOTH N AND Z ON A - 0 RESULT
065034  012704  043200          HH11:   MOV     #043200,R4              ;SET FIV, FIV, AND FD

065040  170104                          LDFPS   R4
065042  012700  065400                  MOV     #HHP8,R0                ;SET AC0 OPERAND
065046  172410                          LDD     (R0),AC0
065050  012700  065410                  MOV     #HHP9,R0                ;FSPC
```

```
065054  172010          HH12:   ADDD    (R0),AC0                ;TEST INSTRUCTION
065056  170205                  STFPS   R5                      ;GET FPS
065060  012700  065270          MOV     #HHDAT0,R0              ;GET THE RESULT
065064  174010                  STD     AC0,(R0)
065066  012701  065370          MOV     #HHP7,R1                ;IS IT CORRECT
065072  012702  000004          MOV     #4,R2
065076  022021          HH13:   CMP     (R0)+,(R1)+
065100  001401                  BEQ     HH16
065102  104000                  EMT                     ;
065104  077204          HH16:   SOB     R2,HH13
065106  052704  100014          BIS     #100014,R4              ;FPS CORRECT?
065112  020405                  CMP     R4,R5
065114  001401                  BEQ     HH17
065116  104000                  EMT                     ;
                        ;TEST THAT CC ARE CLEARED BY R\T
065120  012704  000200  HH17:   MOV     #00200,R4               ;SET FIV, FIV, AND FD
065124  170104                  LDFPS   R4
065126  012700  065400          MOV     #HHP8,R0                ;SET AC0 OPERAND
065132  172410                  LDD     (R0),AC0
065134  012700  065400          MOV     #HHP8,R0                ;FSPC
065140  172010          HH18:   ADDD    (R0),AC0                ;TEST INSTRUCTION
065142  170205                  STFPS   R5                      ;GET FPS
065144  012700  065270          MOV     #HHDAT0,R0              ;GET THE RESULT
065150  174010                  STD     AC0,(R0)
065152  012701  065420          MOV     #HHP10,R1               ;IS IT CORRECT
065156  012702  000004          MOV     #4,R2
065162  022021          HH19:   CMP     (R0)+,(R1)+
065164  001401                  BEQ     HH20
065166  104000                  EMT                     ;
065170  077204          HH20:   SOB     R2,HH19
065172  052704  000000          BIS     #00000,R4               ;FPS CORRECT?
065176  020405                  CMP     R4,R5
065200  001401                  BEQ     HH21
065202  104000                  EMT                     ;
                        ;TEST THAT N IS SET BY R\T
065204  012704  003200  HH21:   MOV     #3200,R4                ;SET FIV, FIV, AND FD
065210  170104                  LDFPS   R4
065212  012700  065350          MOV     #HHP5,R0                ;SET AC0 OPERAND
065216  172410                  LDD     (R0),AC0
065220  012700  065350          MOV     #HHP5,R0                ;FSPC
065224  172010          HH22:   ADDD    (R0),AC0                ;TEST INSTRUCTION
065226  170205                  STFPS   R5                      ;GET FPS
065230  012700  065270          MOV     #HHDAT0,R0              ;GET THE RESULT
065234  174010                  STD     AC0,(R0)
065236  012701  065430          MOV     #HHP11,R1               ;IS IT CORRECT
065242  012702  000004          MOV     #4,R2
065246  022021          HH23:   CMP     (R0)+,(R1)+
065250  001401                  BEQ     HH24
065252  104000                  EMT                     ;
065254  077204          HH24:   SOB     R2,HH23
065256  052704  000010          BIS     #10,R4
065262  020405                  CMP     R4,R5                   ;FPS CORRECT?
065264  001465                  BEQ     HHDONE
065266  104000                  EMT                     ;
065270  000000          HHDAT0: 0
065272  000000                  0
```

```
065274 000000                 0
065276 000000                 0
065300 000452         HHP0:   452
065302 125252                 125252
065304 125252                 125252
065306 125253                 125253
065310 000252         HHP1:   252
065312 125252                 125252
065314 125252                 125252
065316 125252                 125252
065320 000600         HHP2:   600                              ;HHP0 + HHP1 WITH
065322 000000                 0                                ;PROPER NORMALIZATION
065324 000000                 0
065326 000000                 0
065330 000400         HHP3:   400                              ;HHP0 + HHP1 WITH
065332 000000                 0                                ;BAD NORMALIZATION
065334 000000                 0
065336 000000                 0
065340 000000         HHP4:   0
065342 000000                 0
065344 000000                 0
065346 000000                 0
065350 100200         HHP5:   100200
065352 000000                 0
065354 000000                 0
065356 000000                 0
065360 000300         HHP6:   300
065362 000000                 0
065364 000000                 0
065366 000000                 0
065370 100000         HHP7:   100000                           ;HHP7 = HHP8 + HHP9
065372 000000                 0                                ;     = HHP5 + HHP6
065374 000000                 0
065376 000000                 0
065400 000200         HHP8:   200
065402 000000                 0
065404 000000                 0
065406 000000                 0
065410 100300         HHP9:   100300
065412 000000                 0
065414 000000                 0
065416 000000                 0
065420 000400         HHP10:  400                              ;HHP10 = HHP8 + HHP8
065422 000000                 0
065424 000000                 0
065426 000000                 0
065430 100400         HHP11:  100400                           ;HHP11 = HHP5 + HHP5
065432 000000                 0
065434 000000                 0
065436 000000                 0
065440                HHDONE:
065440 004767 033030          JSR     PC,.RSET                 ;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 TYPFD CONTROL G?).
```

```



                                  ;*******************************************************************
                                  ;TEST 457       OVER\UNDER TEST
                                  ;*******************************************************************
065444                            TS457:

                                  ;TEST OVERFLOW CONDITION WITH TRAP DISABLER FIV=0
065444  012704  000200                    MOV     #200,R4           ;CLEAR FIU, FIV, AND SET FD
065450  170104                            LDFPS   R4
065452  012737  065570  000244            MOV     #GGERO,@#FPVECT
065460  012700  066372                    MOV     #GGP5,R0          ;SET AC0 OPERAND
065464  172410                            LDD     (R0),AC0
065466  012700  066372                    MOV     #GGP5,R0          ;FSRC
065472  172010                    GG2:    ADDD    (R0),AC0          ;TEST INSTRUCTION
065474  170205                            STFPS   R5                ;GET FPS
065476  012700  066322                    MOV     #GGDATO,R0        ;GET THE RESULT
065502  174010                            STD     AC0,(R0)
065504  012701  066402                    MOV     #GGP6,R1          ;IS IT CORRECT
065510  012702  000004                    MOV     #4,R2
065514  022021                    GG3:    CMP     (R0)+,(R1)+
065516  001401                            BEQ     GG4
065520  104000                            EMT                       ;
065522  077204                    GG4:    SOB     R2,GG3
065524  052704  000006                    BIS     #6,R4             ;FPS CORRECT?
065530  020405                            CMP     R4,R5
065532  001401                            BEQ     GG5
065534  104000                            EMT                       ;
                                          ;TEST OVERFLOW WITH TRAPS ENABLED
                                          ;FIV = 1
065536  012704  001200            GG5:    MOV     #1200,R4          ;CLEAR FIU, SET FIV, AND FD
065542  170104                            LDFPS   R4
065544  012737  065572  000244            MOV     #GG7,@#FPVECT
065552  012700  066372                    MOV     #GGP5,R0          ;SET AC0 OPERAND
065556  172410                            LDD     (R0),AC0
065560  012700  066372                    MOV     #GGP5,R0          ;FSPC
065564  172010                    GG6:    ADDD    (R0),AC0          ;TEST INSTRUCTION
065566  170000                            CFCC                      ;NO OVERFLOW TRAP OCCURED
065570                            GGERO:
065570  104000                            EMT                       ;
065572  012703  065566            GG7:    MOV     #GG6+2,R3
065576  020316                            CMP     R3,(SP)           ;CHECK STACK DATA
065600  001401                            BEQ     1$
065602  104000                            EMT                       ;
065604  022626                    1$:     CMP     (SP)+,(SP)+
065606  170205                            STFPS   R5
065610  012700  066322                    MOV     #GGDATO,R0        ;GET THE RESULT
065614  174010                            STD     AC0,(R0)
065616  012701  066402                    MOV     #GGP6,R1          ;IS IT CORRECT
065622  012702  000004                    MOV     #4,R2
065626  022021                    GG8:    CMP     (R0)+,(R1)+
065630  001401                            BEQ     GG9
065632  104000                            EMT                       ;
065634  077204                    GG9:    SOB     R2,GG8
065636  052704  100006                    BIS     #100006,R4        ;EXACT ZERO RESULTED IF OVERFLOW
065642  020405                            CMP     R4,R5             ;FPS CORRECT?, CHECK FER, FZ, FV
```

```
065644  001401                          BEQ     1$
065646  104000                          EMT                     ;
065650  012704  000010          1$:     MOV     #10,R4
                                ;CHECK FEC
065654  170305                          STST    R5
065656  020405                          CMP     R4,R5
065660  001401                          BEQ     GG10
065662  104000                          EMT                     ;
                                ;CHECK UNDER FLOW CONDITION WITH
                                ;TRAPS DISABLED (FIU = 0)
065664  012704  000200          GG10:   MOV     #0200,R4        ;SET FIU, FIV, AND FD
065670  170104                          LDFPS   R4
065672  012737  065570  000244          MOV     #GGERO,@#FPVECT
065700  012700  066342                  MOV     #GGP2,R0        ;SET AC0 OPERAND
065704  172410                          LDD     (R0),AC0        ;FSRC
065706  012700  066352                  MOV     #GGP3,R0
065712  172010                  GG11:   ADDD    (R0),AC0        ;TEST INSTRUCTION
065714  170205                          STFPS   R5              ;GET FPS
065716  012700  066322                  MOV     #GGDAT0,R0      ;GET THE RESULT
065722  174010                          STD     AC0,(R0)
065724  012701  066402                  MOV     #GGP6,R1        ;IS IT CORRECT
065730  012702  000004                  MOV     #4,R2
065734  022021                  GG12:   CMP     (R0)+,(R1)+
065736  001401                          BEQ     GG13
065740  104000                          EMT                     ;
065742  077204                  GG13:   SOB     R2,GG12
065744  052704  000004                  BIS     #4,R4           ;FPS CORRECT?
065750  020405                          CMP     R4,R5
065752  001401                          BEQ     GG14
065754  104000                          EMT                     ;
                                        ;CHECK UNDERFLOW CONDITION WITH
                                        ;TRAP ENABLED (FIU = 1)
065756  012704  002200          GG14:   MOV     #2200,R4        ;SET FIU, FIV, AND FD
065762  170104                          LDFPS   R4
065764  012737  066010  000244          MOV     #GG16,@#FPVECT
065772  012700  066342                  MOV     #GGP2,R0        ;SET AC0 OPERAND
065776  172410                          LDD     (R0),AC0        ;FSPC
066000  012700  066352                  MOV     #GGP3,R0
066004  172010                  GG15:   ADDD    (R0),AC0        ;TEST INSTRUCTION
066006  170000                          CFCC
066010  012703  066006          GG16:   MOV     #GG15+2,R3
066014  021603                          CMP     (SP),R3
066016  001401                          BEQ     1$
066020  104000                          EMT                     ;
066022  022626                  1$:     CMP     (SP)+,(SP)+
066024  170205                          STFPS   R5              ;GET FPS
066026  012700  066322                  MOV     #GGDAT0,R0      ;GET THE RESULT
066032  174010                          STD     AC0,(R0)
066034  012701  066412                  MOV     #GGP7,R1        ;IS IT CORRECT
066040  012702  000004                  MOV     #4,R2
066044  022021                  GG17:   CMP     (R0)+,(R1)+
066046  001401                          BEQ     GG18
066050  104000                          EMT                     ;
066052  077204                  GG18:   SOB     R2,GG17
066054  052704  100000                  BIS     #100000,R4
066060  020405                          CMP     R4,R5           ;FPS CORRECT?
```

```
066062  001401                          BEQ     1$
066064  104000                          EMT                     ;
066066  012704  000012          1$:     MOV     #12,R4
                                ;CHECK FEC
066072  170305                          STST    R5
066074  020405                          CMP     R4,R5
066076  001401                          BEQ     GG19
066100  104000                          EMT                     ;
                                ;CHECK UNDERFLOW CONDITION WITH TRAPS
                                ;DISABLED (FIU = 0)
066102  012704  000200          GG19:   MOV     #0200,R4        ;SET FIU, FIV, AND FD
066106  170104                          LDFPS   R4
066110  012737  066226  000244          MOV     #GGER14,@#FPVECT
066116  012700  066342                  MOV     #GGP2,R0        ;SET AC0 OPERAND
066122  172410                          LDD     (R0),AC0
066124  012700  066422                  MOV     #GGF8,R0        ;FSPC
066130  172010                  GG20:   ADDD    (R0),AC0        ;TEST INSTRUCTION
066132  170205                          STFPS   R5              ;GET FPS
066134  012700  066322                  MOV     #GGDAT0,R0      ;GET THE RESULT
066140  174010                          STD     AC0,(R0)
066142  012701  066402                  MOV     #GGP6,R1        ;IS IT CORRECT
066146  012702  000004                  MOV     #4,R2
066152  022021                  GG21:   CMP     (R0)+,(R1)+
066154  001401                          BEQ     GG22
066156  104000                          EMT                     ;
066160  077204                  GG22:   SOB     R2,GG21
066162  052704  000004                  BIS     #4,R4           ;FPS CORRECT?
066166  020405                          CMP     R4,R5
066170  001401                          BEQ     GG23
066172  104000                          EMT                     ;
                                        ;CHECK UNDERFLOW CONDITION WITH TRAP
                                        ;ENABLED (FIU = 1)
066174  012704  002200          GG23:   MOV     #2200,R4        ;SET FIU, FIV, AND FD
066200  170104                          LDFPS   R4
066202  012737  066230  000244          MOV     #GG25,@#FPVECT
066210  012700  066342                  MOV     #GGP2,R0        ;SET AC0 OPERAND
066214  172410                          LDD     (R0),AC0
066216  012700  066422                  MOV     #GGP8,R0        ;FSRC
066222  172010                  GG24:   ADDD    (R0),AC0        ;TEST INSTRUCTION
066224  170000                          CFCC
066226                          GGER14:
066226  104000                          EMT                     ;
066230  012703  066224          GG25:   MOV     #GG24+2,R3
066234  020316                          CMP     R3,(SP)
066236  001401                          BEQ     1$
066240  104000                          EMT                     ;
066242  022626                  1$:     CMP     (SP)+,(SP)+
066244  170205                          STFPS   R5              ;GET FPS
066246  012700  066322                  MOV     #GGDAT0,R0      ;GET THE RESULT
066252  174010                          STD     AC0,(R0)
066254  012701  066432                  MOV     #GGP9,R1        ;IS IT CORRECT
066260  012702  000004                  MOV     #4,R2
066264  022021                  GG26:   CMP     (R0)+,(R1)+
066266  001401                          BEQ     GG27
066270  104000                          EMT                     ;
066272  077204                  GG27:   SOB     R2,GG26
```

```
066274  052704  100004           BIS     #100004,R4
066300  020405                   CMP     R4,R5           ;FPS CORRECT?
066302  001401                   BEQ     1$
066304  104000                   EMT                     ;
066306  012704  000012   1$:     MOV     #12,R4
                                 ;CHECK FEC
066312  170305                   STST    R5
066314  020405                   CMP     R4,R5
066316  001451                   BEQ     GGDONE
066320  104000                   EMT                     ;
066322  000000           GGDATO: 0
066324  000000                   0
066326  000000                   0
066330  000000                   0

066332  000300           GGP1:   300
066334  000000                   0
066336  000000                   0
066340  000000                   0
066342  100200           GGP2:   100200
066344  000000                   0
066346  000000                   0
066350  000000                   0
066352  000200           GGP3:   200
066354  000000                   0
066356  000000                   0
066360  000001                   1
066362  010200           GGP4:   10200
066364  000000                   0
066366  000000                   0
066370  000000                   0
066372  077600           GGP5:   77600                   ;OVER FLOW = GGP5 + GGP5
066374  000000                   0
066376  000000                   0
066400  000000                   0
066402  000000           GGP6:   0                       ;OVERFLOW RESULT
066404  000000                   0                       ;UNDERFLOW RESULT
066406  000000                   0                       ;GGP6 = GGP4 + GGP5
066410  000000                   0                       ;      = GGP3 + GGP2 (FIU = 0)
                                                         ;      = GGP3 + GGP1
066412  062400           GGP7:   62400                   ;GGP7 = GGP3 + GGP2 (FIU = 1)
066414  000000                   0
066416  000000                   0
066420  000000                   0
066422  000340           GGP8:   340
066424  000000                   0
066426  000000                   0
066430  000000                   0
066432  000100           GGP9:   100
066434  000000                   0
066436  000000                   0
066440  000000                   0
066442                   GGDONE:
066442  004767  032026           JSR     PC,.RSET        ;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?).


                        ;*******************************************************************************
                        ;TEST 460        LDCFD AND LDCDF TEST
                        ;*******************************************************************************
066446                  TS460:
                        ;TEST FOR CORRECT AUTO INCREMENT CONSTANT.
066446  012704  000200          MOV     #200,R4         ;SET LONG INTEGER MODE
066452  170104                  LDFPS   R4
066454  012700  067204          MOV     #HXP1,R0
066460  172410                  LDD     (R0),AC0
066462  012700  067214          MOV     #HXP2,R0
066466  177420          HX2:    LDCFD   (R0)+,AC0
066470  020027  067220          CMP     R0,#HXP2+4      ;IS R0 CORRECT
066474  001401                  BEQ     HX3
066476  104000                  EMT                     ;
066500                  HX3:
066500  170205                  STFPS   R5              ;GET FPS
066502  012700  067174          MOV     #HXDAT0,R0
066506  174010                  STD     AC0,(R0)        ;GET AC0
066510  012701  067264          MOV     #HXP7,R1        ;SEE IF RESULT IS
066514  012702  000004          MOV     #4,R2           ;CORRECT
066520  022120          HX4:    CMP     (R1)+,(R0)+
066522  001401                  BEQ     HX7
066524  104000                  EMT                     ;
066526  077204          HX7:    SOB     R2,HX4
066530  012704  000200          MOV     #200,R4         ;FPS CORRECT?
066534  020405                  CMP     R4,R5
066536  001401                  BEQ     HX8
066540  104000                  EMT                     ;
                        ;NOW    TEST    LDCDF
066542  012704  000200  HX8:    MOV #200,R4
066546  170104                  LDFPS R4

066550  012700  067204          MOV     #HXP1,R0
066554  172410                  LDD     (R0),AC0

066556  012700  067214          MOV     #HXP2,R0
066562  170001                  SETF
066564  177420          HX9:    LDCDF   (R0)+,AC0       ;TEST INSTRUCTION
066566  020027  067224          CMP     R0,#HXP2+10     ;WAS A GOOD
066572  001401                  BEQ     HX10
066574  104000                  EMT                     ;

066576                  HX10:
066576  170205                  STFPS   R5
066600  012700  067174          MOV     #HXDAT0,R0
066604  170011                  SETD
066606  174010                  STD     AC0,(R0)        ;GET RESULT
066610  012701  067274          MOV     #HXP8,R1
066614  012702  000004          MOV     #4,R2
066620  022120          HX11:   CMP     (R1)+,(R0)+     ;IS IT CORRECT?
066622  001401                  BEQ     HX14
066624  104000                  EMT                     ;
```

```
066626 077204                  HX14:   SOB     R2,HX11

066630 012704 000000                   MOV     #0,R4           ;FPS CORRECT?
066634 020405                          CMP     R4,R5
066636 001401                          BEQ     HX15
066640 104000                          EMT                     ;
                               ;TEST GR7 IMMEDIATE MODE CONSTANT
066642 012704 000200           HX15:   MOV     #200,R4
066646 170104                          LDFPS   R4              ;SET FD
066650 012737 066700 000004            MOV     #HXER9,@#ERRVECT
066656 005001                          CLR     R1
066660 177427 043243           HX16:   LDCFD   #5201,AC0
066664 005201                  HX165:  INC     R1
066666 005201                          INC     R1
066670 005201                          INC     R1
066672 020127 000003                   CMP     R1,#3           ;SEE IF PC WAS
066676 001401                          BEQ     HX17
066700                         HXER9:
066700 104000                          EMT                     ;
066702 012704 000200           HX17:   MOV     #200,R4
066706 170104                          LDFPS   R4
066710 012700 067254                   MOV     #HXP6,R0
066714 172410                          LDD     (R0),AC0
066716 012700 067214                   MOV     #HXP2,R0
066722 177410                  HX18:   LDCFD   (R0),AC0

066724 012700 067174                   MOV     #HXDAT0,R0
066730 174010                          STD     AC0,(R0)        ;GET RESULT.
066732 012701 067264                   MOV     #HXP7,R1
066736 012702 000004                   MOV     #4,R2
066742 022021                  HX19:   CMP     (R0)+,(R1)+     ;IS RESULT CORRECT?
066744 001401                          BEQ     HX20
066746 104000                          EMT                     ;
066750 077204                  HX20:   SOB     R2,HX19

                               ;TEST LDCFD WITH NEGATIVE OPERAND
066752 012704 000200                   MOV     #200,R4
066756 170104                          LDFPS   R4
066760 012700 067254                   MOV     #HXP6,R0
066764 172410                          LDD     (R0),AC0
066766 012700 067234                   MOV     #HXP4,R0
066772 177410                  HX22:   LDCFD   (R0),AC0

066774 012700 067174                   MOV     #HXDAT0,R0
067000 174010                          STD     AC0,(R0)        ;GET RESULT

067002 012701 067244                   MOV     #HXP5,R1
067006 012702 000004                   MOV     #4,R2
067012 022120                  HX23:   CMP     (R1)+,(R0)+
067014 001401                          BEQ     HX26
067016 104000                          EMT                     ;
067020 077204                  HX26:   SOB     R2,HX23

                               ;TEST   LDCFD   0

067022 012704 000200                   MOV     #200,R4
```

```
067026  170104                   LDFPS   R4

067030  012700  067204           MOV     #HXP1,R0
067034  172410                   LDD     (R0),AC0
067036  172010                   ADDD    (R0),AC0

067040  012700  067204           MOV     #HXP1,R0
067044  177410          HX28:    LDCFD   (R0),AC0

067046  170205                   STFPS   R5

067050  012700  067174           MOV     #HXDAT0,R0
067054  174010                   STD     AC0,(R0)        ;GET RESULT

067056  012701  067204           MOV     #HXP1,R1
067062  012702  000004           MOV     #4,R2
067066  022120          HX29:    CMP     (R1)+,(R0)+     ;IS IT 0?
067070  001401                   BEQ     HX30
067072  104000                   EMT                     ;
067074  077204          HX30:    SOB     R2,HX29

067076  012704  000204           MOV     #204,R4         ;FPS CORRECT
067102  020405                   CMP     R4,R5
067104  001401                   BEQ     HX31
067106  104000                   EMT                     ;
                        ;TEST    LDCFD   0
067110  012704  000200  HX31:    MOV     #200,R4
067114  170104                   LDFPS   R4
067116  012700  067254           MOV     #HXP6,R0
067122  172410                   LDD     (R0),AC0
067124  012700  067204           MOV     #HXP1,R0
067130  177410          HX32:    LDCFD   (R0),AC0
067132  170205                   STFPS   R5
067134  012700  067174           MOV     #HXDAT0,R0
067140  174010                   STD     AC0,(R0)        ;GET RESULT
067142  012701  067204           MOV     #HXP1,R1
067146  012702  000004           MOV     #4,R2
067152  022120          HX33:    CMP     (R1)+,(R0)+     ;IS IT ZERO?
067154  001401                   BEQ     HX34
067156  104000                   EMT                     ;
067160  077204          HX34:    SOB     R2,HX33

067162  012704  000204           MOV     #204,R4         ;FPS CORRECT?
067166  020405                   CMP     R4,R5
067170  001445                   BEQ     HXDONE
067172  104000                   EMT                     ;

067174  000000          HXDAT0:  0
067176  000000                   0
067200  000000                   0
067202  000000                   0

067204  000000          HXP1:    0
067206  000000                   0
067210  000000                   0
067212  000000                   0
```

```

067214  000577                          HXP2:   577
067216  177776                                  177776
067220  177777                                  177777
067222  177776                                  177776
067224  005201                          HXP3:   5201
067226  000000                                  0
067230  000000                                  0
067232  000000                                  0
067234  100577                          HXP4:   100577
067236  177776                                  177776
067240  177777                                  177777
067242  177776                                  177776
067244  100577                          HXP5:   100577
067246  177776                                  177776
067250  000000                                  0
067252  000000                                  0
067254  000252                          HXP6:   252
067256  125252                                  125252
067260  125252                                  125252
067262  125252                                  125252

067264  000577                          HXP7:   577
067266  177776                                  177776
067270  000000                                  0
067272  000000                                  0
067274  000577                          HXP8:   577
067276  177777                                  177777
067300  000000                                  0
067302  000000                                  0

067304                                  HXDONE:
067304  004767  031164                          JSR     PC,.RSET        ;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?).





                                        ;*******************************************************************************
                                        ;TEST 461       CMPD TEST
                                        ;*******************************************************************************
067310                                  TS461:

                                        ;TEST THE CMPD INSTRUCTION WITH (FSRC=AC=0)
067310  004737  070004                  AAA1:   JSR     PC,@#CMPSUB
067314  000000  000000  000000          1$:     .WORD   0,0,0,0                 ;AC0
067322  000000
067324  000000  000000  000000          2$:     .WORD   0,0,0,0                 ;FSRC
067332  000000
067334  000200                          3$:     200                             ;FPS BEFORE EXECUTION
067336  000204                                  204                             ;FPS AFTER EXECUTION


```

```
                                          ;TEST CMPD WITH (AC=0) AND FSRC POSITIVE.
067340  004737  070004                    AAA2:   JSR     PC,@#CMPSUB
067344  000000  000000  000000            1$:     .WORD   0,0,0,0                  ;AC
067352  000000
067354  025252                            2$:     25252                            ;FSRC
067356  052525                                    52525
067360  125252                                    125252
067362  052525                                    52525
067364  000200                            3$:     200                              ;FPS BEFORE EXECUTION
067366  000200                                    200                              ;FPS AFTER EXECUTION

                                          ;TEST CMPD WITH (AC=0) AND FSRC NEGATIVE
067370  004737  070004                    AAA3:   JSR     PC,@#CMPSUB
067374  000000  000000  000000            1$:     .WORD   0,0,0,0                  ;AC
067402  000000
067404  125252                            2$:     125252                           ;FSRC
067406  125252                                    125252
067410  052525                                    52525
067412  125252                                    125252
067414  000200                            3$:     200                              ;FPS BEFORE EXECUTION
067416  000210                                    210                              ;FPS AFTER EXECUTION

                                          ;TEST CMPD WITH (FSRC=0) AND AC POSITIVE
067420  004737  070004                    AAA4:   JSR     PC,@#CMPSUB
067424  025252                            1$:     25252                            ;AC
067426  052525                                    52525
067430  125252                                    125252
067432  052525                                    52525
067434  000000  000000  000000            2$:     .WORD   0,0,0,0                  ;FSRC
067442  000000
067444  000200                            3$:     200                              ;FPS BEFORE EXECUTION
067446  000210                                    210                              ;FPS AFTER EXECUTION


                                          ;TEST CMPD WITH (FSRC=0) AND AC NEGATIVE
067450  004737  070004                    AAA5:   JSR     PC,@#CMPSUB
067454  125252                            1$:     125252                           ;AC
067456  125252                                    125252
067460  052525                                    52525
067462  125252                                    125252
067464  000000  000000  000000            2$:     .WORD   0,0,0,0                  ;FSRC
067472  000000
067474  000200                            3$:     200                              ;FPS BEFORE EXECUTION
067476  000200                                    200                              ;FPS AFTER EXECUTION

                                          ;TEST CMPD WITH AC POSITIVE AND FSRC NEGATIVE
067500  004737  070004                    AAA6:   JSR     PC,@#CMPSUB
067504  052525                            1$:     52525                            ;AC
067506  125252                                    125252
067510  052525                                    52525
067512  125252                                    125252
067514  125252                            2$:     125252                           ;;FSRC
067516  052525                                    52525
067520  125252                                    125252
067522  052525                                    52525
067524  000200                            3$:     200                              ;FPS BEFORE EXECUTION
```

```
067526  000210                  210                           ;FPS AFTER EXECUTION


                        ;TEST CMPD WITH AC NEGATIVE AND FSRC POSITIVE
067530  004737  070004  AAA7:   JSR     PC,@#CMPSUB
067534  125252          1$:     125252                        ;AC
067536  052525                  52525
067540  125252                  125252
067542  052525                  52525
067544  052525          2$:     52525                         ;FSRC
067546  125252                  125252
067550  052525                  52525
067552  125252                  125252
067554  000200          3$:     200                           ;FPS BEFORE EXECUTION
067556  000200                  200                           ;FPS AFTER EXECUTION

                        ;TEST CMPD WITH AC POSITIVE AND FSRC POSITIVE
                        ;AND EAC LESS THAN EFSRC.
067560  004737  070004  AAA8:   JSR     PC,@#CMPSUB
067564  012345          1$:     12345                         ;AC
067566  067654                  67654
067570  032101                  32101
067572  023456                  23456
067574  023456          2$:     23456                         ;FSRC
067576  076543                  76543
067600  021012                  21012
067602  034567                  34567
067604  000200          3$:     200                           ;FPS BEFORE EXECUTION
067606  000200                  200                           ;FPS AFTER EXECUTION


                        ;TEST CMPD WITH AC POSITIVE, FSRC POSITIVE AND EAC GREATER THAN EFSRC
067610  004737  070004  AAA9:   JSR     PC,@#CMPSUB
067614  045676          1$:     45676                         ;AC
067616  054321                  54321
067620  012345                  12345
067622  067654                  67654
067624  034567          2$:     34567                         ;FSRC
067626  065432                  65432
067630  101234                  101234
067632  056765                  56765
067634  000200          3$:     200                           ;FPS BEFORE EXECUTION
067636  000210                  210                           ;FPS AFTER EXECUTION

                        ;TEST CMPD WITH AC POSITIVE, FSRC POSITIVE AND AC ECUAL TO FSRC
067640  004737  070004  AAA10:  JSR     PC,@#CMPSUB
067644  012345          1$:     12345                         ;AC
067646  067012                  67012
067650  034567                  34567
067652  012345                  012345
067654  012345          2$:     12345                         ;FSRC
067656  067012                  67012
067660  034567                  34567
067662  012345                  012345
067664  000200          3$:     200                           ;FPS BEFORE EXECUTION
067666  000204                  204                           ;FPS AFTER EXECUTION
```

```

                                  ;TEST CMPD WITH AC POSITIVE, FSRC POSITIVE, EAC EQUAL TO EFSRC,
                                  ;AND FSRC GREATER THAN AC.
067670  004737  070004            AAA11:  JSR     PC,@#CMPSUB
067674  012345                    1$:     12345                          ;AC
067676  067012                            67012
067700  034567                            34567
067702  012345                            012345
067704  012345                    2$:     12345                          ;FSRC
067706  070123                            70123
067710  045670                            45670
067712  123456                            123456
067714  000200                    3$:     200                            ;FPS BEFORE EXECUTION
067716  000200                            200                            ;FPS AFTER EXECUTION

                                  ;TEST CMPD WITH AC POSITIVE, FSRC POSITIVE, EAC EQUAL TO EFSRC,
                                  ;AND AC GREATER THAN FSRC.
067720  004737  070004            AAA12:  JSR     PC,@#CMPSUB
067724  054321                    1$:     54321                          ;AC
067726  076543                            76543
067730  021076                            21076
067732  054321                            54321
067734  054321                    2$:     54321                          ;FSRC
067736  065432                            65432
067740  107654                            107654
067742  032107                            32107
067744  000200                    3$:     200                            ;FPS BEFORE EXECUTION
067746  000210                            210                            ;FPS AFTER EXECUTION

                                  ;TEST CMPD WITH AC NEGATIVE, FSRC NEGATIVE, EAC EQUAL TO EFSRC,
                                  ;AND AC GREATER THAN FSRC
067750  004737  070004            AAA13:  JSR     PC,@#CMPSUB
067754  112345                    1$:     112345                         ;AC
067756  043210                            43210
067760  076543                            76543
067762  021076                            21076
067764  112345                    2$:     112345                         ;FSRC
067766  054321                            54321
067770  007654                            07654
067772  032107                            32107
067774  000200                    3$:     200                            ;FPS BEFORE EXECUTION
067776  000210                            210                            ;FPS AFTER EXECUTION


070000  000137  070110                    JMP     @#AAADONE              ;FINISHED CMPD TEST.


                                  ;THIS SUBROUTINE, CMPSUB, IS CALLED TO SET UP, EXECUTE
                                  ;AND CHECK THE RESULTS OF A CMPD INSTRUCTION.
                                  ;IT IS CALLED THUS:
                                  ;
                                  ;               JSR     PC,@#CMPSUB
                                  ;       ACARG:  .WORD   X,X,X,X                  ;AC OPERAND
                                  ;       FSRCARG:.WORD   X,X,X,X                  ;FSRC OPERAND
                                  ;       FPSB:   .WORD   X                        ;FPS BEFORE EXECUTION
                                  ;       FPSA:   .WORD   X                        ;FPS AFTER EXECUTION
```

```
                                        ;       FPSE:   .WORD   X                       ;ERROR FPS
                                        ;       ERR:    ERROR   X                       ;FPS ERROR
                                        ;       CONT:                                   ;RETURN ADDRESS
                                        ;
                                        ;THE OPERANDS ARE SET UP (USING AC0 FOR THE AC OPERAND). THEN
                                        ;FPSB IS LOADED INTO THE FPS. THE INSTRUCTION, CMPD, IS EXECUTED.
                                        ;AFTER THE EXECUTION THE FPS IS CHECKED AGAINST FPSA. IF IT IS A MATCH
                                        ;THEN THERE WAS NO ERROR AND CONTROL IS RETURNED TO CONT. IF
                                        ;THE FPS IS INCORRECT IT IS COMPARED WITH FPSE IN AN ATTEMPT TO ANALYSE
                                        ;THE FAILURE. IF THE FPS IS THE SAME AS FPSE THEN CONTROL IS
                                        ;RETURNED TO THE ERROR CALL AT LOCATION ERR. IF THE FPS WAS
                                        ;NOT CORRECT BUT DIDN'T MATCH FPSE A GENERAL ERROR IS REPORTED
                                        ;AND CONTROL IS PASSED TO CONT.

070004  012601                          CMPSUB: MOV     (SP)+,R1                ;PICK UP A POINTER TO THE
                                                                                ;ARGUMENTS.
070006  016100  000020                          MOV     20(R1),R0               ;GET THE FPS BEFORE EXECUTION.
070012  170100                                  LDFPS   R0                      ;LOAD IT INTO THE FPS.

070014  010100                                  MOV     R1,R0                   ;GET ADDRESS OF AC OPERAND.
070016  172410                                  LDD     (R0),AC0                ;LOAD AC0 OPERAND

070020  010100                                  MOV     R1,R0                   ;COMPUTE FSRC OPERAND
070022  062700  000010                          ADD     #10,R0                  ;ADDRESS

070026  000240                                  NOP                             ;FOR SCOPING.
070030  173410                          1$:     CMPD    (R0),AC0                ;EXECUTE THE TEST INSTRUCTION.

070032  170205                                  STFPS   R5                      ;SAVE FPS AFTER INSTRUCTION.

070034  016104  000022                          MOV     22(R1),R4               ;GET EXPECTED FPS.
070040  020405                                  CMP     R4,R5                   ;WAS FPS CORRECT?
070042  020405                                  CMP     R4,R5                   ;WAS FPS CORRECT?
070044  001401                                  BEQ     3$
070046  104000                                  EMT                     ;
070050  012700  070100                  3$:     MOV     #CMPTMP,R0              ;IF FPS WAS CORRECT MAKE SURE
070054  174010                                  STD     AC0,(R0)                ;AC0 WAS NOT AFFECTED BY CMPD.
070056  010102                                  MOV     R1,R2
070060  012703  000004                          MOV     #4,R3
070064  022220                          4$:     CMP     (R2)+,(R0)+
070066  001401                                  BEQ     5$
070070  104000                                  EMT                     ;
070072  077304                          5$:     SOB     R3,4$

070074  000161  000024                          JMP     24(R1)                  ;RETURN

070100  000000  000000  000000          CMPTMP: .WORD   0,0,0,0
070106  000000



070110                                  AAADONE:
070110  004767  030360                          JSR     PC,.RSET        ;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?).
                                   ;*******************************************************************************
                                   ;TEST 462        DIVD WITH (FSRC=0) AND (BUT FD) TEST
                                   ;*******************************************************************************
070114                             TS462:
070114  012704  040200             BBB0:   MOV     #40200,R4       ;SET UP FPS
                                                                   ;WITH INTERRUPTS
                                                                   ;DISABLED.
070120  170104                             LDFPS   R4
070122  012737  070160  000244             MOV     #BBBER1,@#FPVECT;SET UP FOR ANY FP INTERRUPTS.
070130  012700  070344                     MOV     #BBBP1,R0       ;SET UP AC0 = 0
070134  172410                             LDD     (R0),AC0
070136  012701  070344                     MOV     #BBBP1,R1       ;FSRC = 0
070142  174411                     BBB1:   DIVD    (R1),AC0        ;TEST INSTRUCTION
070144  170205                             STFPS   R5              ;GET FPS
070146  170303                             STST    R3              ;GET FEC
070150  012704  140204                     MOV     #140204,R4      ;EXPECTED FPS.
070154  020405                             CMP     R4,R5           ;IS FPS CORRECT.
070156  001401                             BEQ     BBB7
070160                             BBBER1:
070160  104000                             EMT                     ;
070162  012702  000004             BBB7:   MOV     #4,R2           ;EXPECTED FEC.
070166  020203                             CMP     R2,R3           ;IS FEC CORRECT?
070170  001401                             BEQ     BBB2
070172  104000                             EMT                     ;
                                   ;TEST DIVD WITH (FSRC=0) AND TRAPS DISABLED.
070174  012704  040200             BBB2:   MOV     #40200,R4       ;LOAD FPS WITH TRAPS DISABLED.
070200  170104                             LDFPS   R4
070202  012700  070354                     MOV     #BBBP2,R0       ;SET UP AC0 OPERAND (NON ZERO).
070206  172410                             LDD     (R0),AC0
070210  012700  070344                     MOV     #BBBP1,R0       ;FSRC=0
070214  174410                     BBB3:   DIVD    (R0),AC0
070216  170205                             STFPS   R5              ;GET FPS.
070220  170303                             STST    R3              ;GET FEC.
070222  012704  140200                     MOV     #140200,R4              ;EXPECTED FPS.
070226  020405                             CMP     R4,R5           ;IS FPS CORRECT?
070230  001401                             BEQ     1$
070232  104000                             EMT                     ;
070234  012702  000004             1$:     MOV     #4,R2           ;EXPECTED FEC.
070240  020203                             CMP     R2,R3           ;WAS FEC CORRECT?
070242  001401                             BEQ     BBB4
070244  104000                             EMT                     ;
                                   ;TEST DIVD WITH FSRC=0) AND TRAPS ENABLED.
```

```
070246  012704  000200           BBB4:   MOV     #200,R4          ;SET UP FPS. TRAP ENABLED.
070252  170104                           LDFPS   R4

070254  012700  070354                   MOV     #BBBP2,R0        ;SET UP AC0 OPERAND (NON ZERO).
070260  172410                           LDD     (R0),AC0

070262  012737  070302  000244           MOV     #BBB6,@#FPVECT   ;SET UP FOR THE EXPECTED INTERRUPT.
070270  012700  070344                   MOV     #BBBP1,R0        ;FSRC=0

070274  174410                   BBB5:   DIVD    (R0),AC0         ;TEST INSTRUCTION (SHOULD RESULT IN TRAP).
070276  170000                           CFCC
070300  104000                           EMT                      ;
070302  022716  070276           BBB6:   CMP     #BBB5+2,(SP)     ;TRAP TO HERE WHEN THE DIVISION BY 0
                                                                  ;OCCURS. FIRST SEE IF THE ADDRESS OF
                                                                  ;THE TRAP IS 2+THE ADDRESS OF THE TEST
                                                                  ;DIVD INSTRUCTION.
070306  001401                           BEQ     1$
070310  104000                           EMT                      ;
070312  170205                   1$:     STFPS   R5               ;GET FPS.
070314  170303                           STST    R3               ;GET FEC.
070316  022626                           CMP     (SP)+,(SP)+      ;RESET THE STACK.

070320  012704  100200                   MOV     #100200,R4       ;EXPECTED FPS.
070324  020405                           CMP     R4,R5            ;IS FPS CORRECT?
070326  001401                           BEQ     2$
070330  104000                           EMT                      ;
070332  012702  000004           2$:     MOV     #4,R2            ;EXPECTED FEC.
070336  020203                           CMP     R2,R3            ;IS FEC CORRECT?
070340  001411                           BEQ     BBBDONE
070342  104000                           EMT                      ;

070344  000000  000000  000000   BBBP1:  .WORD   0,0,0,0
070352  000000
070354  012345  054321  023456   BBBP2:  .WORD   12345,54321,23456,76543
070362  076543



070364                           BBBDONE:
070364  004767  030104                   JSR     PC,.RSET         ;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?).



                                 ;*******************************************************************
                                 ;TEST 463        DIVF TEST
                                 ;*******************************************************************
070370                           TS463:

                                 ;CHECK DIVF WITH (AC=0).
070370  004767  000404           CCC1:   JSR     PC,DIVFSUB
070374  000000  000000           1$:     .WORD   0,0              ;AC
070400  012345  067012           2$:     .WORD   12345,67012      ;FSRC
```

```
070404  000000  000000      3$:     .WORD   0,0             ;RES
070410  000000              4$:     0                       ;FPS BEFORE EXECUTION
070412  000004                      4                       ;FPS AFTER EXECUTION

                            ;TEST DIVF WITH AC POSITIVE, FSRC POSITIVE AND IN ROUND MODE.
070414  004737  071000      CCC2:   JSR     PC,@#DIVFSUB
070420  065652  125252      1$:     .WORD   65652,125252    ;AC
070424  065600  000000      2$:     .WORD   65600,0         ;FSRC
070430  040252  125252      3$:     .WORD   40252,125252    ;RES
070434  003000              4$:     3000                    ;FPS BEFORE EXECUTION.
070436  003000                      3000                    ;FPS AFTER EXECUTION.

                            ;TEST DIVF WITH AC POSITIVE, FSRC POSITIVE.
070440  004767  000334      CCC3:   JSR     PC,DIVFSUB
070444  076400  000000      1$:     .WORD   76400,0         ;AC
070450  076400  000000      2$:     .WORD   76400,0         ;FSRC
070454  040200  000000      3$:     .WORD   40200,0         ;RES
070460  001000              4$:     1000                    ;FPS BEFORE EXECUTION.
070462  001000                      1000                    ;FPS AFTER EXECUTION.
                            ;TEST DIVF WITH BOTH OPERANDS POSITIVE.
070464  004737  071000      CCC4:   JSR     PC,@#DIVFSUB
070470  056777  177777      1$:     .WORD   56777,177777    ;AC
070474  054200  000000      2$:     .WORD   54200,0         ;FSRC
070500  042777  177777      3$:     .WORD   42777,177777    ;RES
070504  000000              4$:     0                       ;FPS BEFORE EXECUTION.
070506  000000                      0                       ;FPS AFTER EXECUTION.

                            ;TEST THE DIVF INSTRUCTION:
070510  004737  071000      CCC5:   JSR     PC,@#DIVFSUB
070514  012377  177777      1$:     .WORD   12377,177777    ;AC
070520  012300  000000      2$:     .WORD   12300,0         ;FSRC
070524  040252  125252      3$:     .WORD   40252,125252    ;RES
070530  000000              4$:     0                       ;FPS BEFORE EXECUTION.
070532  000000                      0                       ;FPS AFTER EXECUTION.

                            ;TEST DIVIDE ALGORITHM. TEST ROUND CONSTANT.
070534  004737  071000      CCC6:   JSR     PC,@#DIVFSUB
070540  064600  000001      1$:     .WORD   64600,1         ;AC
070544  066600  000000      2$:     .WORD   66600,0         ;FSRC
070550  036200  000001      3$:     .WORD   36200,1         ;RES
070554  000000              4$:     0                       ;FPS BEFORE EXECUTION.
070556  000000                      0                       ;FPS AFTER EXECUTION.

                            ;TEST DIVF.
070560  004737  071000      CCC7:   JSR     PC,@#DIVFSUB
070564  034577  177776      1$:     .WORD   34577,177776    ;AC
070570  023400  000000      2$:     .WORD   23400,0         ;FSRC
070574  051377  177776      3$:     .WORD   51377,177776    ;RES
070600  000017              4$:     17                      ;FPS BEFORE EXECUTION.
070602  000000                      0                       ;FPS AFTER EXECUTION.


                            ;DIVF TEST.
070604  004737  071000      CCC8:   JSR     PC,@#DIVFSUB
070610  067652  125252      1$:     .WORD   67652,125252    ;AC
070614  056500  000000      2$:     .WORD   56500,0         ;FSRC
```

```
070620 051343 107070    3$:     .WORD   51343,107070    ;RES
070624 000000           4$:     0                       ;FPS BEFORE EXECUTION.
070626 000000                   0                       ;FPS AFTER EXECUTION.

                        ;DIVF WITH AC NEGATIVE, FSRC NEGATIVE.
070630 004737 071000    CCC9:   JSR     PC,@#DIVFSUB
070634 140400 000000    1$:     .WORD   140400,0        ;AC
070640 140500 000000    2$:     .WORD   140500,0        ;FSRC
070644 040052 125253    3$:     .WORD   040052,125253   ;RES
070650 000000           4$:     0                       ;FPS BEFORE EXECUTION.
070652 000000                   0                       ;FPS AFTER EXECUTION.

                        ;DIVF WITH AC NEGATIVE AND FSRC POSITIVE.
070654 004737 071000    CCC10:  JSR     PC,@#DIVFSUB
070660 160077 000000    1$:     .WORD   160077,0        ;AC
070664 040277 000000    2$:     .WORD   40277,0         ;FSRC
070670 160000 000000    3$:     .WORD   160000,0        ;RES
070674 000007           4$:     7                       ;FPS BEFORE EXECUTION.
070676 000010                   10                      ;FPS AFTER EXECUTION.

                        ;DIVF WITH AC POSITIVE AND FSRC NEGATIVE.
070700 004737 071000    CCC11:  JSR     PC,@#DIVFSUB
070704 040400 000000    1$:     .WORD   40400,0         ;AC
070710 140500 000000    2$:     .WORD   140500,0        ;FSRC
070714 140052 125253    3$:     .WORD   140052,125253   ;RES
070720 000017           4$:     17                      ;FPS BEFORE EXECUTION.
070722 000010                   10                      ;FPS AFTER EXECUTION.


                        ;TEST DIVF BOTH OPERANDS POSITIVE AND TRUNCATE MODE.
070724 004737 071000    CCC12:  JSR     PC,@#DIVFSUB
070730 060100 000001    1$:     .WORD   60100,1         ;AC
070734 040300 000000    2$:     .WORD   40300,0         ;FSRC
070740 060000 000000    3$:     .WORD   60000,0         ;RES
070744 000052           4$:     52                      ;FPS BEFORE EXECUTION.
070746 000040                   40                      ;FPS AFTER EXECUTION.

                        ;DIVF WITH POSITIVE OPERANDS AND ROUND MODE.
070750 004767 000024    CCC13:  JSR     PC,DIVFSUB
070754 060100 000001    1$:     .WORD   60100,1         ;AC
070760 040300 000000    2$:     .WORD   40300,0         ;FSRC
070764 060000 000001    3$:     .WORD   60000,1         ;RES
070770 000005           4$:     5                       ;FPS BEFORE EXECUTION.
070772 000000                   0                       ;FPS AFTER EXECUTION.

070774 000137 071116            JMP     @#CCCDONE       ;GO TO NEXT TEST.

                        ;THIS SUBROUTINE, DIVFSUB, IS CALLED TO SET UP, EXECUTE
                        ;AND CHECK THE RESULT OF A DIVF INSTRUCTION. IT IS CALLED THUS:
                        ;
                        ;               JSR     PC,@#DIVFSUB
                        ;       ACARG:  .WORD   X,X             ;AC OPERAND
                        ;       FSRCARG:.WORD   X,X             ;FSRC OPERAND
                        ;       RES:    .WORD   X,X             ;EXPECTED RESULT
                        ;       FPSB:   .WORD   X               ;FPS BEFORE EXECUTION
                        ;       FPSA:   .WORD   X               ;FPS AFTER EXECUTION
```

```
                                        ;       ERRES:  .WORD   X,X                     ;ERROR RESULT
                                        ;       ERR:    ERROR   X                       ;RESULT ERROR
                                        ;       CONT:                                   ;RETURN ADDRESS
                                        ;
                                        ;THE OPERANDS ARE SET UP (USING AC0 FOR THE AC OPERAND). THEN
                                        ;FPSB IS LOADED INTO THE FPS. THE INSTRUCTION, DIVF IS EXECUTED.
                                        ;AFTER THE EXECUTION THE RESULT IS CHECKED AGAINST THE
                                        ;EXPECTED CORRECT RESULT, RES. IF IT IS CORRECT THEN THE FPS
                                        ;IS CHECKED WITH THE EXPECTED CORRECT FPS, FPSA. IF THE FPS WAS
                                        ;INCORRECT THEN IT IS REPORTED. IF THE RESULT WAS INCORRECT IT
                                        ;IS COMPARED WITH ERRES IN AN ATTEMPT TO ANALYSE THE ERROR. IF
                                        ;THE INCORRECT RESULT MATCHED ERRES THEN CONTROL IS PASSED TO
                                        ;THE ERROR CALL AT ERR. IF THE INCORRECT RESULT DID NOT MATCH ERRES
                                        ;THEN THE FAILURE IS REPORTED IN DIVFSUB AND CONTROL IS PASSED TO
                                        ;CONT. IF NO ERRORS ARE DETECTED THEN DIVFSUB RETURNS CONTROL
                                        ;TO CONT.

 071000  012601                         DIVFSUB:        MOV     (SP)+,R1        ;GET A POINTER TO THE ARGUMENTS.
 071002  012700  000200                         MOV     #200,R0         ;SET FD MODE.
 071006  170100                                 LDFPS   R0
 071010  010100                                 MOV     R1,R0           ;LOAD THE AC OPERAND.
 071012  172410                                 LDD     (R0),AC0
 071014  016100  000014                         MOV     14(R1),R0       ;LOAD THE FPS
 071020  170100                                 LDFPS   R0
 071022  010100                                 MOV     R1,R0
 071024  062700  000004                         ADD     #4,R0           ;ESTABLISH A POINTER TO FSRC.

 071030  174410                         1$:     DIVF    (R0),AC0        ;TEST INSTRUCTION.

 071032  170204                                 STFPS   R4              ;GET THE FPS.
 071034  012700  000200                         MOV     #200,R0         ;SET FD MODE
 071040  170100                                 LDFPS   R0

 071042  012700  071106                         MOV     #DIVFT,R0       ;GET THE RESULT OF THE DIVF.
 071046  174010                                 STD     AC0,(R0)
 071050  021061  000010                         CMP     (R0),10(R1)     ;IS THE RESULT CORRECT?
 071054  001401                                 BEQ     2$
 071056  104000                                 EMT                     ;
 071060  026061  000002  000012         2$:     CMP     2(R0),12(R1)
 071066  001401                                 BEQ     3$
 071070  104000                                 EMT                     ;
 071072  026104  000016                 3$:     CMP     16(R1),R4       ;IS FPS CORRECT?
 071076  001401                                 BEQ     4$
 071100  104000                                 EMT                     ;
 071102  000161  000020                 4$:     JMP     20(R1)          ;IF NO ERRORS OCCURRED RETURN.

 071106  000000  000000  000000         DIVFT:  .WORD   0,0,0,0
 071114  000000

 071116                                 CCCDONE:
 071116  004767  027352                         JSR     PC,.RSET        ;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?).
```

```


                                          ;*****************************************************************************
                                          ;TEST 464        DIVD TEST
                                          ;*****************************************************************************
071122                                    TS464:

                                          ;DIVD TEST WITH POSITIVE OPERANDS AND IN ROUND MODE.
071122  004737  071466                    DDD1:   JSR     PC,@#DIVDSUB
071126  034277  000000  000000            1$:     .WORD   34277,0,0,0        ;AC
071134  000000
071136  040277  000000  000000            2$:     .WORD   40277,0,0,0        ;FSRC
071144  000000
071146  034200  000000  000000            3$:     .WORD   34200,0,0,0        ;RES
071154  000000
071156  000200                            4$:     200                        ;FPS BEFORE EXECUTION.
071160  000200                                    200                        ;FPS AFTER EXECUTION.

                                          ;DIVD WITH AC NEGATIVE AND FSRC POSITIVE IN TRUNCATE MODE.
071162  004737  071466                    DDD2:   JSR     PC,@#DIVDSUB
071166  134277  000000  000000            1$:     .WORD   134277,0,0,0       ;AC
071174  000000
071176  040277  000000  000000            2$:     .WORD   40277,0,0,0        ;FSRC
071204  000000
071206  134200  000000  000000            3$:     .WORD   134200,0,0,0       ;RES
071214  000000
071216  000207                            4$:     207                        ;FPS BEFORE EXECUTION.
071220  000210                                    210                        ;FPS AFTER EXECUTION.

                                          ;DIVD TEST WITH OPERANDS BOTH NEGATIVE AND IN TRUNCATE MODE.
071222  004767  000240                    DDD3:   JSR     PC,DIVDSUB
071226  134300  000000  000000            1$:     .WORD   134300,0,0,1       ;AC
071234  000001
071236  140300  000000  000000            2$:     .WORD   140300,0,0,0       ;FSRC
071244  000000
071246  034200  000000  000000            3$:     .WORD   34200,0,0,0        ;RES
071254  000000
071256  000250                            4$:     250                        ;FPS BEFORE EXECUTION.
071260  000240                                    240                        ;FPS AFTER EXECUTION.

                                          ;DIVD WITH AC POSITIVE AND FSRC NEGATIVE IN ROUND MODE.
071262  004737  071466                    DDD4:   JSR     PC,@#DIVDSUB
071266  034300  000000  000000            1$:     .WORD   34300,0,0,1        ;AC
071274  000001
071276  140300  000000  000000            2$:     .WORD   140300,0,0,0       ;FSRC
071304  000000
071306  134200  000000  000000            3$:     .WORD   134200,0,0,1       ;RES
071314  000001
071316  000207                            4$:     207                        ;FPS BEFORE EXECUTION.
071320  000210                                    210                        ;FPS AFTER EXECUTION.

                                          ;DIVD TEST.
071322  004737  071466                    DDD5:   JSR     PC,@#DIVDSUB
071326  100400  000000  000000            1$:     .WORD   100400,0,0,0       ;AC
071334  000000
071336  000500  000000  000000            2$:     .WORD   500,0,0,0          ;FSRC
```

```
071344  000000
071346  140052  125252              3$:     .WORD   140052,125252   ;RES
071352  125252  125252                      .WORD   125252,125252
071356  007647                      4$:     7647                    ;FPS BEFORE EXECUTION.
071360  007650                              7650                    ;FPS AFTER EXECUTION.


                                    ;DIVD TEST WITH AC POSITIVE AND FSRC NEGATIVE IN ROUND MODE.
071362  004737  071466              DDD6:   JSR     PC,@#DIVDSUB
071366  000400  000000  000000      1$:     .WORD   400,0,0,0       ;AC
071374  000000
071376  100500  000000  000000      2$:     .WORD   100500,0,0,0    ;FSRC
071404  000000
071406  140052  125252              3$:     .WORD   140052,125252   ;RES
071412  125252  125253                      .WORD   125252,125253
071416  007707                      4$:     7707                    ;FPS BEFORE EXECUTION.
071420  007710                              7710                    ;FPS AFTER EXECUTION.

                                    ;DIVD TEST.
071422  004737  071466              DDD7:   JSR     PC,@#DIVDSUB
071426  170360  170360              1$:     .WORD   170360,170360   ;AC
071432  170360  170360                      .WORD   170360,170360
071436  170360  170360              2$:     .WORD   170360,170360   ;FSRC
071442  170360  170360                      .WORD   170360,170360
071446  040200  000000  000000      3$:     .WORD   40200,0,0,0     ;RES
071454  000000
071456  007717                      4$:     7717                    ;FPS BEFORE EXECUTION.
071460  007700                              7700                    ;FPS AFTER EXECUTION.

071462  000137  071610                      JMP     @#DDDDONE       ;GO TO NEXT TEST.


                                    ;THIS SUBROUTINE, DIVDSUB, IS CALLED TO SET UP, EXECUTE
                                    ;AND CHECK THE RESULT OF A DIVD INSTRUCTION. IT IS CALLED THUS:
                                    ;
                                    ;               JSR     PC,@#DIVDSUB
                                    ;       ACARG:  .WORD   X,X,X,X                 ;AC OPERAND
                                    ;       FSRCARG:.WORD   X,X,X,X                 ;FSRC OPERAND
                                    ;       RES:    .WORD   X,X,X,X                 ;EXPECTED RESULT
                                    ;       FPSB:   .WORD   X                       ;FPS BEFORE EXECUTION
                                    ;       FPSA:   .WORD   X                       ;FPS AFTER EXECUTION
                                    ;       ERRES:  .WORD   X,X,X,X                 ;ERROR RESULT
                                    ;       ERR:    ERROR   X                       ;RESULT ERROR
                                    ;       CONT:                                   ;RETURN ADDRESS
                                    ;
                                    ;THE OPERANDS ARE SET UP (USING AC0 FOR THE AC OPERAND). THEN
                                    ;FPSB IS LOADED INTO THE FPS. THE INSTRUCTION, DIVD IS EXECUTED.
                                    ;AFTER THE EXECUTION THE RESULT IS CHECKED AGAINST THE
                                    ;EXPECTED CORRECT RESULT, RES. IF IT IS CORRECT THEN THE FPS
                                    ;IS CHECKED WITH THE EXPECTED CORRECT FPS, FPSA. IF THE FPS WAS
                                    ;INCORRECT THEN IT IS REPORTED. IF THE RESULT WAS INCORRECT IT
                                    ;IS COMPARED WITH ERRES IN AN ATTEMPT TO ANALYSE THE ERROR. IF
                                    ;THE INCORRECT RESULT MATCHED ERRES THEN CONTROL IS PASSED TO
                                    ;THE ERROR CALL AT ERR. IF THE INCORRECT RESULT DID NOT MATCH ERRES
                                    ;THEN THE FAILURE IS REPORTED IN DIVDSUB AND CONTROL IS PASSED TO
                                    ;CONT. IF NO ERRORS ARE DETECTED THEN DIVDSUB RETURNS CONTROL
```

```
                                  ;TO CONT.

071466  012601                    DIVDSUB:        MOV     (SP)+,R1        ;GET A POINTER TO THE ARGUMENTS.
071470  012700  000200                    MOV     #200,R0         ;SET FD MODE.
071474  170100                            LDFPS   R0

071476  010100                            MOV     R1,R0           ;SET UP THE AC0 OPERAND.
071500  172410                            LDD     (R0),AC0
071502  016100  000030                    MOV     30(R1),R0       ;LOAD THE FPS.
071506  170100                            LDFPS   R0

071510  010100                            MOV     R1,R0           ;ESTABLISH A POINTER TO FSRC.
071512  062700  000010                    ADD     #10,R0

071516  174410                    1$:     DIVD    (R0),AC0        ;EXECUTE THE TEST INSTRUCTION.
071520  170204                            STFPS   R4              ;GET THE FPS.
071522  012700  000200                    MOV     #200,R0         ;SET FD MODE.
071526  170100                            LDFPS   R0
071530  012700  071600                    MOV     #DIVDT,R0       ;GET THE RESULT.
071534  174010                            STD     AC0,(R0)
071536  010102                            MOV     R1,R2           ;CHECK THE RESULT.
071540  062702  000020                    ADD     #20,R2
071544  012703  071600                    MOV     #DIVDT,R3
071550  012705  000004                    MOV     #4,R5
071554  022223                    2$:     CMP     (R2)+,(R3)+
071556  001401                            BEQ     3$
071560  104000                            EMT                     ;
071562  077504                    3$:     SOB     R5,2$

071564  026104  000032                    CMP     32(R1),R4       ;IS FPS CORRECT?
071570  001401                            BEQ     4$
071572  104000                            EMT                     ;
071574  000161  000034            4$:     JMP     34(R1)          ;RETURN.
071600  000000  000000  000000    DIVDT:  .WORD   0,0,0,0
071606  000000

071610                            DDDDONE:
071610  004767  026660                    JSR     PC,.RSET        ;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?).




                                  ;*****************************************************************
                                  ;TEST 465        MULF TEST
                                  ;*****************************************************************
071614                            TS465:

                                  ;MULF WITH (FSRC=AC=0)
071614  004737  072224            EEE1:   JSR     PC,@#MULFSUB
071620  000000  000000            1$:     .WORD   0,0             ;AC
071624  000000  000000            2$:     .WORD   0,0             ;FSRC
071630  000000  000000            3$:     .WORD   0,0             ;RES
071634  007517                    4$:     7517                    ;FPS BEFORE EXECUTION.
```

```
071636  007504                   7504                    ;FPS AFTER EXECUTION.

                         ;MULF WITH (FSRC=0).
071640  004737  072224   EEE2:   JSR     PC,@#MULFSUB
071644  071625  034435   1$:     .WORD   71625,34435     ;AC
071650  000000  000000   2$:     .WORD   0,0             ;FSRC
071654  000000  000000   3$:     .WORD   0,0             ;RES
071660  000013           4$:     13                      ;FPS BEFORE EXECUTION.
071662  000004                   4                       ;FPS AFTER EXECUTION.

                         ;MULF WITH (AC=0)
071664  004737  072224   EEE3:   JSR     PC,@#MULFSUB
071670  000000  000000   1$:     .WORD   0,0             ;AC
071674  071625  153443   2$:     .WORD   071625,153443   ;FSRC
071700  000000  000000   3$:     .WORD   0,0             ;RES
071704  007500           4$:     7500                    ;FPS BEFORE EXECUTION.
071706  007504                   7504                    ;FPS AFTER EXECUTION.

                         ;MULF WITH AC POSITIVE AND FSRC POSITIVE IN ROUND MODE.
071710  004737  072224   EEE4:   JSR     PC,@#MULFSUB
071714  040200  000000   1$:     .WORD   40200,0         ;AC
071720  040177  177777   2$:     .WORD   40177,-1        ;FSRC
071724  040177  177777   3$:     .WORD   40177,-1        ;RES
071730  000017           4$:     17                      ;FPS BEFORE EXECUTION.
071732  000000                   0                       ;FPS AFTER EXECUTION.

                         ;MULF WITH AC POSITIVE AND FSRC POSITIVE IN TRUNCATE MODE.
071734  004767  000264   EEE5:   JSR     PC,MULFSUB
071740  040177  177777   1$:     .WORD   40177,-1        ;AC
071744  040200  000000   2$:     .WORD   40200,0         ;FSRC
071750  040177  177777   3$:     .WORD   40177,-1        ;RES
071754  000040           4$:     40                      ;FPS BEFORE EXECUTION.
071756  000040                   40                      ;FPS AFTER EXECUTION.

                         ;MULF WITH BOTH OPERANDS POSITIVE NORMALIZE TEST.
071760  004737  072224   EEE6:   JSR     PC,@#MULFSUB
071764  040100  000000   1$:     .WORD   40100,0         ;AC
071770  040100  000000   2$:     .WORD   40100,0         ;FSRC
071774  040020  000000   3$:     .WORD   40020,0         ;RES
072000  000012           4$:     12                      ;FPS BEFORE EXECUTION.
072002  000000                   0                       ;FPS AFTER EXECUTION.

                         ;MULF WITH BOTH OPERANDS POSITIVE IN ROUND MODE.
072004  004737  072224   EEE7:   JSR     PC,@#MULFSUB
072010  017500  000000   1$:     .WORD   17500,0         ;AC
072014  023652  125252   2$:     .WORD   23652,125252    ;FSRC
072020  003177  177777   3$:     .WORD   3177,-1         ;RES
072024  007417           4$:     7417                    ;FPS BEFORE EXECUTION.
072026  007400                   7400                    ;FPS AFTER EXECUTION.

                         ;MULF WITH AC POSITIVE AND FSRC NEGATIVE IN ROUND MODE.
072030  004737  072224   EEE8:   JSR     PC,@#MULFSUB
072034  040342  000000   1$:     .WORD   40342,0         ;AC
072040  176542  000000   2$:     .WORD   176542,0        ;FSRC
072044  176707  102000   3$:     .WORD   176707,102000   ;RES
072050  000007           4$:     7                       ;FPS BEFORE EXECUTION.
```

```
072052  000010                          10                      ;FPS AFTER EXECUTION.

                                ;MULF WITH AC NEGATIVE AND FSRC POSITIVE IN ROUND MODE.
072054  004737  072224          EEE9:   JSR     PC,@#MULFSUB
072060  140200  000000          1$:     .WORD   140200,0        ;AC
072064  007417  007417          2$:     .WORD   7417,7417       ;FSRC
072070  107417  007417          3$:     .WORD   107417,7417     ;RES
072074  000000                  4$:     0                       ;FPS BEFORE EXECUTION.
072076  000010                          10                      ;FPS AFTER EXECUTION.

                                ;MULF WITH BOTH OPERANDS NEGATIVE IN ROUND MODE.
072100  004737  072224          EEE10:  JSR     PC,@#MULFSUB
072104  144600  000000          1$:     .WORD   144600,0        ;AC
072110  154000  000000          2$:     .WORD   154000,0        ;FSRC
072114  060400  000000          3$:     .WORD   60400,0         ;RES
072120  000017                  4$:     17                      ;FPS BEFORE EXECUTION.
072122  000000                          0                       ;FPS AFTER EXECUTION.

                                ;MULF BOTH OPERANDS NEGATIVE IN ROUND MODE.
072124  004737  072224          EEE11:  JSR     PC,@#MULFSUB
072130  140300  000000          1$:     .WORD   140300,0        ;AC
072134  160000  000001          2$:     .WORD   160000,1        ;FSRC
072140  060100  000002          3$:     .WORD   60100,2         ;RES
072144  000010                  4$:     10                      ;FPS BEFORE EXECUTION.
072146  000000                          0                       ;FPS AFTER EXECUTION.

                                ;MULF WITH AC POSITIVE AND FSRC NEGATIVE IN TRUNCATE MODE.
072150  004737  072224          EEE12:  JSR     PC,@#MULFSUB
072154  060000  000001          1$:     .WORD   60000,1         ;AC
072160  140300  000000          2$:     .WORD   140300,0        ;FSRC
072164  160100  000001          3$:     .WORD   160100,1        ;RES
072170  007547                  4$:     7547                    ;FPS BEFORE EXECUTION.
072172  007550                          7550                    ;FPS AFTER EXECUTION.

                                ;MULF WITH AC POSITIVE AND FSRC POSITIVE IN ROUND MODE.
072174  004737  072224          EEE13:  JSR     PC,@#MULFSUB
072200  040277  000000          1$:     .WORD   40277,0         ;AC
072204  060000  000001          2$:     .WORD   60000,1         ;FSRC
072210  060077  000001          3$:     .WORD   60077,1         ;RES
072214  000014                  4$:     14                      ;FPS BEFORE EXECUTION.
072216  000000                          0                       ;FPS AFTER EXECUTION.

072220  000167  000116                  JMP     EEEDONE         ;GO TO THE NEXT TEST.

                                ;THIS SUBROUTINE, MULFSUB, IS CALLED TO SET UP, EXECUTE
                                ;AND CHECK THE RESULT OF A MULF INSTRUCTION. IT IS CALLED THUS:
                                ;
                                ;                  JSR     PC,@#MULFSUB
                                ;       ACARG:     .WORD   X,X                  ;AC OPERAND
                                ;       FSRCARG:.WORD      X,X                  ;FSRC OPERAND
                                ;       RES:       .WORD   X,X                  ;EXPECTED RESULT
                                ;       FPSB:      .WORD   X                    ;FPS BEFORE EXECUTION
                                ;       FPSA:      .WORD   X                    ;FPS AFTER EXECUTION
                                ;       ERRES:     .WORD   X,X                  ;ERROR RESULT
                                ;       ERR:       ERROR   X                    ;RESULT ERROR
                                ;       CONT:                                   ;RETURN ADDRESS
```

```
                                        ;
                                        ;THE OPERANDS ARE SET UP (USING AC0 FOR THE AC OPERAND). THEN
                                        ;FPSB IS LOADED INTO THE FPS. THE INSTRUCTION, MULF IS EXECUTED.
                                        ;AFTER THE EXECUTION THE RESULT IS CHECKED AGAINST THE
                                        ;EXPECTED CORRECT RESULT, RES. IF IT IS CORRECT THEN THE FPS
                                        ;IS CHECKED WITH THE EXPECTED CORRECT FPS, FPSA. IF THE FPS WAS
                                        ;INCORRECT THEN IT IS REPORTED. IF THE RESULT WAS INCORRECT IT
                                        ;IS COMPARED WITH ERRES IN AN ATTEMPT TO ANALYSE THE ERROR. IF
                                        ;THE INCORRECT RESULT MATCHED ERRES THEN CONTROL IS PASSED TO
                                        ;THE ERROR CALL AT ERR. IF THE INCORRECT RESULT DID NOT MATCH ERRES
                                        ;THEN THE FAILURE IS REPORTED IN MULFSUB AND CONTROL IS PASSED TO
                                        ;CONT. IF NO ERRORS ARE DETECTED THEN MULFSUB RETURNS CONTROL
                                        ;TO CONT.

072224  012601                          MULFSUB:        MOV     (SP)+,R1        ;GET A POINTER TO THE ARGUMENTS.
072226  012700  000200                          MOV     #200,R0         ;SET FD MODE.
072232  170100                                  LDFPS   R0
072234  010100                                  MOV     R1,R0           ;LOAD THE AC OPERAND
072236  172410                                  LDD     (R0),AC0
072240  016100  000014                          MOV     14(R1),R0       ;LOAD THE FPS
072244  170100                                  LDFPS   R0
072246  010100                                  MOV     R1,R0
072250  062700  000004                          ADD     #4,R0           ;ESTABLISH A POINTER TO FSRC.

072254  171010                          1$:     MULF    (R0),AC0        ;TEST INSTRUCTION.

072256  170204                                  STFPS   R4              ;GET THE FPS.
072260  012700  000200                          MOV     #200,R0         ;SET FD MODE
072264  170100                                  LDFPS   R0

072266  012700  072332                          MOV     #MULFT,R0       ;GET THE RESULT OF THE MULF.
072272  174010                                  STD     AC0,(R0)
072274  021061  000010                          CMP     (R0),10(R1)     ;IS THE RESULT CORRECT?
072300  001401                                  BEQ     2$
072302  104000                                  EMT                     ;
072304  026061  000002  000012          2$:     CMP     2(R0),12(R1)
072312  001401                                  BEQ     3$
072314  104000                                  EMT                     ;
072316  026104  000016                  3$:     CMP     16(R1),R4       ;IS FPS CORRECT?
072322  001401                                  BEQ     4$
072324  104000                                  EMT                     ;
072326  000161  000020                  4$:     JMP     20(R1)          ;IF NO ERRORS OCCURRED RETURN.

072332  000000  000000  000000          MULFT:  .WORD   0,0,0,0
072340  000000

072342                                  EEEDONE:
072342  004767  026126                          JSR     PC,.RSET        ;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?).




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

```
                                        ;TEST 466       MULD TEST
                                        ;*********************************************************************
072346                                  TS466:

                                        ;MULD TEST WITH AC POSITIVE AND FSRC POSITIVE.
072346  004737  072552                  FFF1:   JSR     PC,@#MULDSUB
072352  040200  000000  000000          1$:     .WORD   40200,0,0,0     ;AC
072360  000000
072362  023777  177777  177777          2$:     .WORD   23777,-1,-1,-1  ;FSRC
072370  177777
072372  023777  177777  177777          3$:     .WORD   23777,-1,-1,-1  ;RES
072400  177777
072402  000217                          4$:     217                     ;FPS BEFORE EXECUTION.
072404  000200                                  200                     ;FPS AFTER ECECUTION.

                                        ;MULD TEST WITH BOTH OPERANDS POSITIVE TRUNCATION TEST.
072406  004767  000140                  FFF2:   JSR     PC,MULDSUB
072412  065400  000000  000000          1$:     .WORD   65400,0,0,1     ;AC
072420  000001
072422  037577  177777  177777          2$:     .WORD   37577,-1,-1,-2  ;FSRC
072430  177776
072432  064777  177777  177777          3$:     .WORD   64777,-1,-1,-1  ;RES
072440  177777
072442  000247                          4$:     247                     ;FPS BEFORE EXECUTION.
072444  000240                                  240                     ;FPS AFTER EXECUTION.

                                        ;MULD TEST WITH BOTH OPERANDS NEGATIVE IN ROUND MODE.
072446  004737  072552                  FFF3:   JSR     PC,@#MULDSUB
072452  137577  177777  177777          1$:     .WORD   137577,-1,-1,-2 ;AC
072460  177776
072462  165400  000000  000000          2$:     .WORD   165400,0,0,1    ;FSRC
072470  000001
072472  065000  000000  000000          3$:     .WORD   65000,0,0,0     ;RES
072500  000000
072502  007717                          4$:     7717                    ;FPS BEFORE EXECUTION.
072504  007700                                  7700                    ;FPS AFTER EXECUTION.

                                        ;MULD   TEST WITH AC POSITIVE AND FSRC NEGATIVE IN ROUND MODE.
072506  004737  072552                  FFF4:   JSR     PC,@#MULDSUB
072512  017500  000000  000000          1$:     .WORD   17500,0,0,0     ;AC
072520  000000
072522  123652  125252                  2$:     .WORD   123652,125252   ;FSRC
072526  125252  125252                          .WORD   125252,125252
072532  103177  177777  177777          3$:     .WORD   103177,-1,-1,-1 ;RES
072540  177777
072542  000200                          4$:     200                     ;FPS BEFORE EXECUTION.
072544  000210                                  210                     ;FPS AFTER EXECUTION.

072546  000167  000122                          JMP     FFFDONE

                                        ;THIS SUBROUTINE, MULDSUB, IS CALLED TO SET UP, EXECUTE
                                        ;AND CHECK THE RESULT OF A MULD INSTRUCTION. IT IS CALLED THUS::
                                        ;                       JSR     PC,@#MULDSUB
                                        ;       ACARG:  .WORD   X,X,X,X                 ;AC OPERAND
                                        ;       FSRCARG:.WORD   X,X,X,X                 ;FSRC OPERAND
                                        ;       RES:    .WORD   X,X,X,X                 ;EXPECTED RESULT
```

```
                                ;       FPSB:   .WORD   X                       ;FPS BEFORE EXECUTION
                                ;       FPSA:   .WORD   X                       ;FPS AFTER EXECUTION
                                ;       ERRES:  .WORD   X,X,X,X                 ;ERROR RESULT
                                ;       ERR:    ERROR   X                       ;RESULT ERROR
                                ;       CONT:                                   ;RETURN ADDRESS
                                ;
                                ;THE OPERANDS ARE SET UP (USING AC0 FOR THE AC OPERAND). THEN
                                ;FPSB IS LOADED INTO THE FPS. THE INSTRUCTION, MULD IS EXECUTED.
                                ;AFTER THE EXECUTION THE RESULT IS CHECKED AGAINST THE
                                ;EXPECTED CORRECT RESULT, RES. IF IT IS CORRECT THEN THE FPS
                                ;IS CHECKED WITH THE EXPECTED CORRECT FPS, FPSA. IF THE FPS WAS
                                ;INCORRECT THEN IT IS REPORTED. IF THE RESULT WAS INCORRECT IT
                                ;IS COMPARED WITH ERRES IN AN ATTEMPT TO ANALYSE THE ERROR. IF
                                ;THE INCORRECT RESULT MATCHED ERRES THEN CONTROL IS PASSED TO
                                ;THE ERROR CALL AT ERR. IF THE INCORRECT RESULT DID NOT MATCH ERRES
                                ;THEN THE FAILURE IS REPORTED IN MULDSUB AND CONTROL IS PASSED TO
                                ;CONT. IF NO ERRORS ARE DETECTED THEN MULDSUB RETURNS CONTROL
                                ;TO CONT.

072552  012601                  MULDSUB:        MOV     (SP)+,R1        ;GET A POINTER TO THE ARGUMENTS.
072554  012700  000200                  MOV     #200,R0         ;SET FD MODE.
072560  170100                          LDFPS   R0

072562  010100                          MOV     R1,R0           ;SET UP THE AC0 OPERAND.
072564  172410                          LDD     (R0),AC0
072566  016100  000030                  MOV     30(R1),R0       ;LOAD THE FPS.
072572  170100                          LDFPS   R0

072574  010100                          MOV     R1,R0           ;ESTABLISH A POINTER TO FSRC.
072576  062700  000010                  ADD     #10,R0

072602  171010                  1$:     MULD    (R0),AC0        ;EXECUTE THE TEST INSTRUCTION.

072604  170204                          STFPS   R4              ;GET THE FPS.
072606  012700  000200                  MOV     #200,R0         ;SET FD MODE.
072612  170100                          LDFPS   R0

072614  012700  072664                  MOV     #MULDT,R0       ;GET THE RESULT.
072620  174010                          STD     AC0,(R0)
072622  010102                          MOV     R1,R2           ;CHECK THE RESULT.
072624  062702  000020                  ADD     #20,R2
072630  012703  072664                  MOV     #MULDT,R3
072634  012705  000004                  MOV     #4,R5
072640  022223                  2$:     CMP     (R2)+,(R3)+
072642  001401                          BEQ     3$
072644  104000                          EMT                     ;
072646  077504                  3$:     SOB     R5,2$

072650  026104  000032                  CMP     32(R1),R4       ;IS FPS CORRECT?
072654  001401                          BEQ     4$
072656  104000                          EMT                     ;
072660  000161  000034          4$:     JMP     34(R1)          ;RETURN.

072664  000000  000000  000000  MULDT:  .WORD   0,0,0,0
072672  000000
072674                          FFFDONE:
```

```
072674  004767  025574          JSR     PC,.RSET        ;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?).




                        ;*************************************************************************
                        ;TEST 467       UNDER\OVER FLOW, USING MULF WITH TRAPS DISABLED, TEST
                        ;*************************************************************************
072700                  TS467:

                        ;UNDERFLOW, WITH EXPONENT OF RESULT = -129
072700  004737  073044  III1:   JSR     PC,@#OVUNFNT
072704  020200  000000  1$:     .WORD   20200,0         ;AC
072710  020000  000000  2$:     .WORD   20000,0         ;FSRC
072714  000000  000000  3$:     .WORD   0,0             ;RES
072720  000000          5$:     0                       ;FPS BEFORE EXECUTION.
072722  000004                  4                       ;FPS AFTER EXECUTION.
072724  000012          6$:     12                      ;FEC
072726  177777                  -1                      ;FLAG

                        ;UNDERFLOW, WITH EXPONENT OF RESULT = -193
072730  004737  073044  III2:   JSR     PC,@#OVUNFNT
072734  010200  000000  1$:     .WORD   10200,0         ;AC
072740  010000  000000  2$:     .WORD   10000,0         ;FSRC
072744  000000  000000  3$:     .WORD   0,0             ;RES
072750  005013          5$:     5013                    ;FPS BEFORE EXECUTION.
072752  005004                  5004                    ;FPS AFTER EXECUTION.
072754  000012          6$:     12                      ;FEC
072756  177777                  -1                      ;FLAG
                        ;OVERFLOW, EXPONENT OF RESULT = 128
072760  004737  073044  III3:   JSR     PC,@#OVUNFNT
072764  060200  000000  1$:     .WORD   60200,0         ;AC
072770  060000  000000  2$:     .WORD   60000,0         ;FSRC
072774  000000  000000  3$:     .WORD   0,0             ;RES
073000  000000          5$:     0                       ;FPS BEFORE EXECUTION.
073002  000006                  6                       ;FPS AFTER EXECUTION.
073004  000010          6$:     10                      ;FEC
073006  000000                  0                       ;FLAG
                        ;OVERFLOW, EXPONENT OF RESULT = 130
073010  004737  073044  III4:   JSR     PC,@#OVUNFNT
073014  060200  000000  1$:     .WORD   60200,0         ;AC
073020  060200  000000  2$:     .WORD   60200,0         ;FSRC
073024  000000  000000  3$:     .WORD   0,0             ;RES
073030  006011          5$:     6011                    ;FPS BEFORE EXECUTION.
073032  006006                  6006                    ;FPS AFTER EXECUTION.
073034  000010          6$:     10                      ;FEC
073036  000000                  0                       ;FLAG
073040  000167  000132  8$:     JMP     IIIDONE         ;GO TO NEXT TEST.



                        ;THIS SUBROUTINE, OVUNFNT, IS USED TO SET UP THE OPERANDS, EXECUTE
                        ;THE MULF INSTRUCTION AND CHECK THE RESULTS OF AN INSTRUCTION WITH
```

```
                                ;OPERANDS WHICH SHOULD RESULT IN EITHER OVERFLOW OR UNDERFLOW. A CALL
                                ;TO IT IS MADE THUS:
                                ;
                                ;               ACARG:  .WORD   X,X             ;AC OPERAND
                                ;               FSRCARG:.WORD   X,X             ;FSRC OPERAND
                                ;               RES:    .WORD   X,X             ;EXPECTED RESULT
                                ;               ERRES:  .WORD   X,X             ;ERROR RESULT
                                ;               FPSB:   .WORD   X               ;FPS BEFORE EXECUTION
                                ;               FPSA:   .WORD   X               ;FPS AFTER EXECUTION
                                ;               FEC:    .WORD   X               ;EXPECTED FEC
                                ;               FLAG:   .WORD   X               ;0/-1,OVER/UNDER FLOW FLAG
                                ;               ERR1:   ERROR   X               ;TRAP ERROR.
                                ;               BR      CONT
                                ;               ERR2:   ERROR   X               ;DATA, RESULT ERROR
                                ;               CONT:                           ;RETURN ADDRESS
                                ;
                                ;THE OPERANDS ARE SET UP (USING AC0 AS THE ACCUMULATOR). THEN
                                ;THE MULF INSTRUCTION IS EXECUTED. IF NO TRAP OCCURS THEN THE
                                ;RESULT IS CHECKED AGAINST RES. IF THE RESULT IS CORRECT THEN THE FPS IS
                                ;COMPARED WITH FPSA IF THIS TOO IS CORRECT OVUNFNT RETURNS CONTROL
                                ;TO THE CALLING ROUTINE AT CONT. IF THE FPS IS BAD OVUNFNT
                                ;REPORTS THIS FAILURE AND THEN RETURNS TO CONT. IF THE RESULT OF THE
                                ;MULF IS INCORRECT, THE INCORRECT RESULT IS COMPARED WITH THE
                                ;ANTICIPATED FAILING DATA PATTERN, ERRES. IF THE FAILURE IN
                                ;THE RESULT WAS ANTICIPATED CORRECTLY TO BE ERRES THEN OVUNFNT
                                ;WILL TRANSFER CONTROL TO THE ERROR CALL AT ERR2. OTHERWISE THE
                                ;RESULT WAS INCORRECT BUT WAS NOT ANTICIPATED AND OVUNFNT WILL
                                ;REPORT THE FAILURE AFTER WHICH CONTROL WILL BE PASSED TO CONT.
                                ;IF A TRAP OCCURS (IT SHOULD NOT) THEN OVUNFNT WILL READ THE FEC.
                                ;SHOULD THE FEC MATCH THE ANTICIPATED FEC OVUNFNT WILL
                                ;STORE ALL DATA AND TRANSFER CONTROL TO THE ERROR CALL AT ERR1. IF THE
                                ;FEC IS NOT THE SAME AS THE ANTICIPATED FEC OVUNFNT WILL REPORT
                                ;THE ERROR AND RETURN TO CONT. NOTE THAT OVUNFNT USES THE FLAG
                                ;TO TELL WHETHER OR NOT THESE PARTICULAR OPERANDS WILL RESULT IN
                                ;UNDERFLOW (FLAG=-1) OR OVERFLOW (FLAG=0).

073044  012601                  OVUNFNT:        MOV     (SP)+,R1        ;GET A POINTER TO THE ARGUMENTS.
073046  012700  000200                  MOV     #200,R0         ;SET FD MODE.
073052  170100                          LDFPS   R0

073054  010100                          MOV     R1,R0           ;LOAD AC0, OPERAND.
073056  172410                          LDD     (R0),AC0
073060  016100  000014                  MOV     14(R1),R0       ;LOAD THE FPS
073064  170100                          LDFPS   R0
073066  012737  073146  000244          MOV     #25$,@#FPVECT   ;SET UP THE FP TRAP VECTOR IN CASE
                                                                ;OF ERROR.
073074  010100                          MOV     R1,R0           ;COMPUTE THE ADDRESS OF FSRC.
073076  062700  000004                  ADD     #4,R0

073102  171010                  1$:     MULF    (R0),AC0        ;TEST INSTRUCTION.

073104  170204                  2$:     STFPS   R4              ;GET FPS.
073106  170305                          STST    R5              ;GET FEC.
073110  012700  000200                  MOV     #200,R0         ;SET FD MODE.
073114  170100                          LDFPS   R0
073116  012700  073166                  MOV     #OVFNTT,R0      ;GET THE RESULT.
```

```
073122  174010                                   STD     AC0,(R0)
073124  012700  073166                           MOV     #OVFNTT,R0      ;CHECK THE RESULT.
073130  010102                                   MOV     R1,R2
073132  062702  000010                           ADD     #10,R2
073136  012703  000002                           MOV     #2,R3
073142  022022                           3$:     CMP     (R0)+,(R2)+
073144  001401                                   BEQ     5$
073146                                   25$:
073146  104000                                   EMT                     ;
073150  077304                           5$:     SOB     R3,3$

073152  026104  000016                           CMP     16(R1),R4       ;WAS FPS CORRECT?
073156  001401                                   BEQ     4$
073160  104000                                   EMT                     ;
073162  000161  000024                   4$:     JMP     24(R1)          ;RETURN, TEST COMPLETED.

073166  000000  000000  000000           OVFNTT: .WORD   0,0,0,0
073174  000000

073176                                   IIIDONE:
073176  004767  025272                           JSR     PC,.RSET        ;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?).





                                         ;*************************************************************************
                                         ;TEST 470       UNDER\OVER FLOW, USING MULD WITH TRAP DISABLED, TEST
                                         ;*************************************************************************
073202                                   TS470:

                                         ;UNDERFLOW, EXPONENT OF RESULT=-129
073202  004737  073426                   JJJ1:   JSR     PC,@#OVUNDNT
073206  020200  000000                   1$:     .WORD   20200,0         ;AC
073212  127272  000000                           .WORD   127272,0
073216  020000  000000  000000           2$:     .WORD   20000,0,0,0     ;FSRC
073224  000000
073226  000000  000000  000000           3$:     .WORD   0,0,0,0         ;RES
073234  000000
073236  000200                           5$:     200                     ;FPS BEFORE EXECUTION.
073240  000204                                   204                     ;FPS AFTER EXECUTION.
073242  000012                           6$:     12                      ;FEC
073244  177777                                   -1                      ;FLAG

                                         ;UNDERFLOW, EXPONENT OF RESULT = -193
073246  004737  073426                   JJJ2:   JSR     PC,@#OVUNDNT
073252  010200  000000                   1$:     .WORD   10200,0         ;AC
073256  123456  000000                           .WORD   123456,0
073262  010000  000000  000000           2$:     .WORD   10000,0,0,0     ;FSRC
073270  000000
073272  000000  000000  000000           3$:     .WORD   0,0,0,0         ;RES
073300  000000
073302  005213                           5$:     5213                    ;FPS BEFORE EXECUTION.
```

```
073304  005204                               5204                    ;FPS AFTER EXECUTION.
073306  000012                       6$:     12                      ;FEC
073310  177777                               -1                      ;FLAG

                                     ;OVERFLOW, EXPONENT OF RESULT = 128
073312  004737  073426               JJJ3:   JSR     PC,@#OVUNDNT
073316  060200  000000               1$:     .WORD   60200,0         ;AC
073322  065432  000000                       .WORD   65432,0
073326  060000  000000  000000       2$:     .WORD   60000,0,0,0     ;FSRC
073334  000000
073336  000000  000000  000000       3$:     .WORD   0,0,0,0         ;RES
073344  000000
073346  000200                       5$:     200                     ;FPS BEFORE EXECUTION.
073350  000206                               206                     ;FPS AFTER EXECUTION.
073352  000010                       6$:     10                      ;FEC
073354  000000                               0                       ;FLAG

                                     ;OVERFLOW, EXPONENT OF RESULT = 130
073356  004737  073426               JJJ4:   JSR     PC,@#OVUNDNT
073362  060200  000000               1$:     .WORD   60200,0         ;AC
073366  125252  000000                       .WORD   125252,0
073372  060200  000000  000000       2$:     .WORD   60200,0,0,0     ;FSRC
073400  000000
073402  000000  000000  000000       3$:     .WORD   0,0,0,0         ;RES
073410  000000
073412  006211                       5$:     6211                    ;FPS BEFORE EXECUTION.
073414  006206                               6206                    ;FPS AFTER EXECUTION.
073416  000010                       6$:     10                      ;FEC
073420  000000                               0                       ;FLAG
073422  000137  073560               8$:     JMP     @#JJJDONE       ;GO TO NEXT TEST

                                     ;THIS SUBROUTINE, OVUNDNT, IS USED TO SET UP THE OPERANDS, EXECUTE
                                     ;THE MULD INSTRUCTION AND CHECK THE RESULTS OF AN INSTRUCTION WITH
                                     ;OPERANDS WHICH SHOULD RESULT IN EITHER OVERFLOW OR UNDERFLOW. A CALL
                                     ;TO IT IS MADE THUS:
                                     ;
                                     ;               ACARG:  .WORD   X,X,X,X         ;AC OPERAND
                                     ;               FSRCARG:.WORD   X,X,X,X         ;FSRC OPERAND
                                     ;               RES:    .WORD   X,X,X,X         ;EXPECTED RESULT
                                     ;               ERRES:  .WORD   X,X,X,X         ;ERROR RESULT
                                     ;               FPSB:   .WORD   X               ;FPS BEFORE EXECUTION
                                     ;               FPSA:   .WORD   X               ;FPS AFTER EXECUTION
                                     ;               FEC:    .WORD   X               ;EXPECTED FEC
                                     ;               FLAG:   .WORD   X               ;0/-1,OVER/UNDER FLOW FLAG
                                     ;               ERR1:   ERROR   X               ;TRAP ERROR.
                                     ;               BR      CONT
                                     ;               ERR2:   ERROR   X               ;DATA, RESULT ERROR
                                     ;               CONT:                           ;RETURN ADDRESS
                                     ;
                                     ;THE OPERANDS ARE SET UP (USING AC0 AS THE ACCUMULATOR). THEN
                                     ;THE MULD INSTRUCTION IS EXECUTED. IF NO TRAP OCCURS THEN THE
                                     ;RESULT IS CHECKED AGAINST RES. IF THE RESULT IS CORRECT THEN THE FPS IS
                                     ;COMPARED WITH FPSA IF THIS TOO IS CORRECT OVUNDNT RETURNS CONTROL
                                     ;TO THE CALLING ROUTINE AT CONT. IF THE FPS IS BAD OVUNDNT
                                     ;REPORTS THIS FAILURE AND THEN RETURNS TO CONT. IF THE RESULT OF THE
                                     ;MULD IS INCORRECT, THE INCORRECT RESULT IS COMPARED WITH THE
```

```
                                        ;ANTICIPATED FAILING DATA PATTERN, ERRES. IF THE FAILURE IN
                                        ;THE RESULT WAS ANTICIPATED CORRECTLY TO BE ERRES THEN OVUNDNT
                                        ;WILL TRANSFER CONTROL TO THE ERROR CALL AT ERR2. OTHERWISE THE
                                        ;RESULT WAS INCORRECT BUT WAS NOT ANTICIPATED AND OVUNDNT WILL
                                        ;REPORT THE FAILURE AFTER WHICH CONTROL WILL BE PASSED TO CONT.
                                        ;IF A TRAP OCCURS (IT SHOULD NOT) THEN OVUNDNT WILL READ THE FEC.
                                        ;SHOULD THE FEC MATCH THE ANTICIPATED FEC OVUNDNT WILL
                                        ;STORE ALL DATA AND TRANSFER CONTROL TO THE ERROR CALL AT ERR1. IF THE
                                        ;FEC IS NOT THE SAME AS THE ANTICIPATED FEC OVUNDNT WILL REPORT
                                        ;THE ERROR AND RETURN TO CONT. NOTE THAT OVUNDNT USES THE FLAG
                                        ;TO TELL WHETHER OR NOT THESE PARTICULAR OPERANDS WILL RESULT IN
                                        ;UNDERFLOW (FLAG=-1) OR OVERFLOW (FLAG=0).

073426  012601                          OVUNDNT:        MOV     (SP)+,R1        ;GET A POINTER TO THE ARGUMENTS.
073430  012700  000200                          MOV     #200,R0         ;SET FD MODE.
073434  170100                                  LDFPS   R0

073436  010100                                  MOV     R1,R0           ;LOAD AC0, OPERAND.
073440  172410                                  LDD     (R0),AC0
073442  016100  000030                          MOV     30(R1),R0       ;LOAD THE FPS.
073446  170100                                  LDFPS   R0
073450  012737  073530  000244                  MOV     #25$,@#FPVECT   ;SET UP THE FP TRAP VECTOR IN CASE
                                                                        ;OF ERROR.
073456  010100                                  MOV     R1,R0           ;COMPUTE THE ADDRESS OF FSRC.
073460  062700  000010                          ADD     #10,R0

073464  171010                          1$:     MULD    (R0),AC0        ;TEST INSTRUCTION.

073466  170204                          2$:     STFPS   R4              ;GET FPS.
073470  170305                                  STST    R5              ;GET FEC.
073472  012700  000200                          MOV     #200,R0         ;SET FD MODE.
073476  170100                                  LDFPS   R0
073500  012700  073550                          MOV     #OVDNTT,R0      ;GET THE RESULT.
073504  174010                                  STD     AC0,(R0)
073506  012700  073550                          MOV     #OVDNTT,R0      ;CHECK THE RESULT.
073512  010102                                  MOV     R1,R2
073514  062702  000020                          ADD     #20,R2
073520  012703  000004                          MOV     #4,R3
073524  022022                          3$:     CMP     (R0)+,(R2)+
073526  001401                                  BEQ     5$
073530                                  25$:
073530  104000                                  EMT                     ;
073532  077304                          5$:     SOB     R3,3$

073534  026104  000032                          CMP     32(R1),R4       ;WAS FPS CORRECT?
073540  001401                                  BEQ     4$
073542  104000                                  EMT                     ;
073544  000161  000040                  4$:     JMP     40(R1)          ;RETURN, TEST COMPLETED.

073550  000000  000000  000000          OVDNTT: .WORD   0,0,0,0
073556  000000

073560                                  JJJDONE:
073560  004767  024710                          JSR     PC,.RSET        ;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?).




                                ;*****************************************************************************
                                ;TEST 471       UNDER\OVER FLOW, USING MULF WITH TRAPS ENABLED, TEST
                                ;*****************************************************************************
073564                          TS471:

                                ;UNDERFLOW, EXPONENT OF RESULT = -129
073564  004737  073730          KKK1:   JSR     PC,@#OVUNFT
073570  020123  045676          1$:     .WORD   20123,45676     ;AC
073574  020200  000000          2$:     .WORD   20200,0         ;FSRC
073600  000123  045676          3$:     .WORD   123,45676       ;RES
073604  002000                  5$:     2000                    ;FPS BEFORE EXECUTION.
073606  102004                          102004                  ;FPS AFTER EXECUTION
073610  000012                  6$:     12                      ;FEC
073612  177777                          -1                      ;FLAG

                                ;UNDERFLOW, EXPONENT OF THE RESULT = -193
073614  004737  073730          KKK3:   JSR     PC,@#OVUNFT
073620  010127  127272          1$:     .WORD   10127,127272    ;AC
073624  010200  000000          2$:     .WORD   10200,0         ;FSRC
073630  060127  127272          3$:     .WORD   60127,127272    ;RES
073634  007017                  5$:     7017                    ;FPS BEFORE EXECUTION.
073636  107000                          107000                  ;FPS AFTER EXECUTION.
073640  000012                  6$:     12                      ;FEC
073642  177777                          -1

                                ;OVERFLOW, EXPONENT OF THE RESULT = 128
073644  004737  073730          KKK4:   JSR     PC,@#OVUNFT
073650  060252  125252          1$:     .WORD   60252,125252    ;AC
073654  060000  000000          2$:     .WORD   60000,0         ;FSRC
073660  000052  125252          3$:     .WORD   000052,125252   ;RES
073664  001000                  5$:     1000                    ;FPS BEFORE EXECUTION.
073666  101006                          101006                  ;FPS AFTER EXECUTION.
073670  000010                  6$:     10                      ;FEC
073672  000000                          0                       ;FLAG

                                ;OVERFLOW, EXPONENT OF RESULT = 130
073674  004737  073730          KKK5:   JSR     PC,@#OVUNFT
073700  060345  067654          1$:     .WORD   60345,67654     ;AC
073704  060200  000000          2$:     .WORD   60200,0         ;FSRC
073710  000345  067654          3$:     .WORD   345,67654       ;RES
073714  007015                  5$:     7015                    ;FPS BEFORE EXECUTION.
073716  107002                          107002                  ;FPS AFTER EXECUTION.
073720  000010                  6$:     10                      ;FEC
073722  000000                          0                       ;FLAG
073724  000167  000162          8$:     JMP     KKKDONE

                                ;THIS SUBROUTINE, OVUNFT, IS USED TO SET UP THE OPERANDS, EXECUTE
                                ;THE MULF INSTRUCTION AND CHECK THE RESULTS OF AN INSTRUCTION WITH
                                ;OPERANDS WHICH SHOULD RESULT IN EITHER OVERFLOW OR UNDERFLOW. A CALL
                                ;TO IT IS MADE THUS:
```

```
                                        ;
                                        ;               ACARG:  .WORD   X,X             ;AC OPERAND
                                        ;               FSRCARG:.WORD   X,X             ;FSRC OPERAND
                                        ;               RES:    .WORD   X,X             ;EXPECTED RESULT
                                        ;               ERRES:  .WORD   X,X             ;ERROR RESULT
                                        ;               FPSB:   .WORD   X               ;FPS BEFORE EXECUTION
                                        ;               FPSA:   .WORD   X               ;FPS AFTER EXECUTION
                                        ;               FEC:    .WORD   X               ;EXPECTED FEC
                                        ;               FLAG:   .WORD   X               ;0/-1,OVER/UNDER FLOW FLAG
                                        ;               ERR1:   ERROR   X               ;TRAP ERROR.
                                        ;               BR      CONT
                                        ;               ERR2:   ERROR   X               ;DATA, RESULT ERROR
                                        ;               CONT:                           ;RETURN ADDRESS
                                        ;
                                        ;THE OPERANDS ARE SET UP (USING AC0 AS THE ACCUMULATOR). THEN
                                        ;THE MULF INSTRUCTION IS EXECUTED. IF THE TRAP OCCURS THEN THE
                                        ;RESULT IS CHECKED AGAINST RES. IF THE RESULT IS CORRECT THEN THE FPS IS
                                        ;COMPARED WITH FPSA IF THIS TOO IS CORRECT OVUNFT RETURNS CONTROL
                                        ;TO THE CALLING ROUTINE AT CONT. IF THE FPS IS BAD OVUNFT
                                        ;REPORTS THIS FAILURE AND THEN RETURNS TO CONT. THE FEC IS TREATED
                                        ;IN THE SAME WAY. IF THE RESULT OF THE
                                        ;MULF IS INCORRECT, THE INCORRECT RESULT IS COMPARED WITH THE
                                        ;ANTICIPATED FAILING DATA PATTERN, ERRES. IF THE FAILURE IN
                                        ;THE RESULT WAS ANTICIPATED CORRECTLY TO BE ERRES THEN OVUNFT
                                        ;WILL TRANSFER CONTROL TO THE ERROR CALL AT ERR2. OTHERWISE THE
                                        ;RESULT WAS INCORRECT BUT WAS NOT ANTICIPATED AND OVUNFT WILL
                                        ;REPORT THE FAILURE AFTER WHICH CONTROL WILL BE PASSED TO CONT.
                                        ;IF NO TRAP OCCURS CONTROL IS PASSED TO ERR1.
                                        ;NOTE THAT OVUNFNT USES THE FLAG
                                        ;TO TELL WHETHER OR NOT THESE PARTICULAR OPERANDS WILL RESULT IN
                                        ;UNDERFLOW (FLAG=-1) OR OVERFLOW (FLAG=0).

073730  012601                          OVUNFT: MOV     (SP)+,R1        ;GET A POINTER TO THE ARGUMENTS.
073732  012700  000200                          MOV     #200,R0         ;SET FD MODE.
073736  170100                                  LDFPS   R0
073740  010100                                  MOV     R1,R0           ;LOAD AC0, OPERAND.
073742  172410                                  LDD     (R0),AC0
073744  016100  000014                          MOV     14(R1),R0       ;LOAD THE FPS.
073750  170100                                  LDFPS   R0
073752  012737  073774  000244                  MOV     #50$,@#FPVECT   ;SET UP THE FP TRAP VECTOR IN CASE
                                                                        ;OF ERROR.
073760  010100                                  MOV     R1,R0           ;COMPUTE THE ADDRESS OF FSRC.
073762  062700  000004                          ADD     #4,R0

073766  171010                          1$:     MULF    (R0),AC0        ;TEST INSTRUCTION. SHOULD CAUSE TRAP.
073770  170000                          2$:     CFCC
073772  104000                                  EMT                     ;
073774  011602                          50$:    MOV     (SP),R2         ;TRAP TO HERE AND SEE IF THE PC OF THE
073776  020227  073770                          CMP     R2,#2$          ;TRAP WAS THAT OF THE MULF INSTRUCTION.
074002  001401                                  BEQ     51$
074004  104000                                  EMT                     ;
074006  022626                          51$:    CMP     (SP)+,(SP)+     ;RESET THE STACK
074010  170204                                  STFPS   R4              ;GET FPS.
074012  170305                                  STST    R5              ;GET FEC.
074014  012700  000200                          MOV     #200,R0         ;SET FD MODE.
074020  170100                                  LDFPS   R0
```

```
074022  012700  074102                          MOV     #OVFTT,R0       ;GET THE RESULT.
074026  174010                                  STD     AC0,(R0)
074030  012700  074102                          MOV     #OVFTT,R0       ;CHECK THE RESULT.
074034  010102                                  MOV     R1,R2
074036  062702  000010                          ADD     #10,R2
074042  012703  000002                          MOV     #2,R3
074046  022022                          3$:     CMP     (R0)+,(R2)+
074050  001401                                  BEQ     5$
074052  104000                                  EMT                     ;
074054  077304                          5$:     SOB     R3,3$

074056  026104  000016                          CMP     16(R1),R4       ;WAS FPS CORRECT?
074062  001401                                  BEQ     6$
074064  104000                                  EMT                     ;
074066  026105  000020                  6$:     CMP     20(R1),R5       ;IS FEC CORRECT?
074072  001401                                  BEQ     4$
074074  104000                                  EMT                     ;
074076  000161  000024                  4$:     JMP     24(R1)          ;RETURN, TEST COMPLETED.

074102  000000  000000  000000          OVFTT:  .WORD   0,0,0,0
074110  000000

074112                                  KKKDONE:
074112  004767  024356                          JSR     PC,.RSET        ;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?).





                                        ;*******************************************************************************
                                        ;TEST 472       UNDER\OVER FLOW, USING MULD WITH TRAPS ENABLED, TEST
                                        ;*******************************************************************************
074116                                  TS472:

                                        ;UNDERFLOW, EXPONENT OF RESULT = -129
074116  004737  074342                  LLL1:   JSR     PC,@#OVUNDT
074122  020052  125252                  1$:     .WORD   20052,125252    ;AC
074126  125252  125252                          .WORD   125252,125252
074132  020300  000000  000000          2$:     .WORD   20300,0,0,0     ;FSRC
074140  000000
074142  000177  177777  177777          3$:     .WORD   177,-1,-1,-1    ;RES
074150  177777
074152  002200                          5$:     2200                    ;FPS BEFORE EXECUTION.
074154  102204                                  102204                  ;FPS AFTER EXECUTION.
074156  000012                          6$:     12                      ;FEC
074160  177777                                  -1                      ;FLAG

                                        ;UNDERFLOW, EXPONENT OF THE RESULT = -193
074162  004737  074342                  LLL2:   JSR     PC,@#OVUNDT
074166  010327  127272                  1$:     .WORD   10327,127272    ;AC
074172  036363  045454                          .WORD   36363,45454
074176  010000  000000  000000          2$:     .WORD   10000,0,0,0     ;FSRC
074204  000000
```

```
074206 060127 127272          3$:     .WORD   60127,127272     ;RES
074212 036363 045454                  .WORD   36363,45454
074216 007217                 5$:     7217                     ;FPS BEFORE EXECUTION.
074220 107200                         107200                   ;FPS AFTER EXECUTION.
074222 000012                 6$:     12                       ;FEC
074224 177777                         -1                       ;FLAG

                              ;OVERFLOW, EXPONENT OF THE RESULT = 128
074226 004737 074342          LLL3:   JSR     PC,@#OVUNDT
074232 060252 125252          1$:     .WORD   60252,125252     ;AC
074236 125252 125252                  .WORD   125252,125252    ;FSRC
074242 160100 000000 000000   2$:     .WORD   160100,0,0,0     ;FSRC
074250 000000
074252 100177 177777 177777   3$:     .WORD   100177,-1,-1,-1  ;RES
074260 177777
074262 001200                 5$:     1200                     ;FPS BEFORE EXECUTION.
074264 101216                         101216                   ;FPS AFTER EXECUTION.
074266 000010                 6$:     10                       ;FEC
074270 000000                         0                        ;FLAG

                              ;OVERFLOW, EXPONENT OF THE RESULT = 130
074272 004737 074342          LLL4:   JSR     PC,@#OVUNDT
074276 060345 067654          1$:     .WORD   60345,67654      ;AC
074302 056765 045676                  .WORD   56765,45676
074306 060200 000000 000000   2$:     .WORD   60200,0,0,0      ;FSRC
074314 000000
074316 000345 067654          3$:     .WORD   345,67654        ;RES
074322 056765 045676                  .WORD   56765,45676
074326 007215                 5$:     7215                     ;FPS BEFORE EXECUTION.
074330 107202                         107202                   ;FPS AFTER EXECUTION.
074332 000010                 6$:     10                       ;FEC
074334 000000                         0                        ;FLAG
074336 000137 074524          8$:     JMP     @#LLLDONE

                              ;THIS SUBROUTINE, OVUNDT, IS USED TO SET UP THE OPERANDS, EXECUTE
                              ;THE MULD INSTRUCTION AND CHECK THE RESULTS OF AN INSTRUCTION WITH
                              ;OPERANDS WHICH SHOULD RESULT IN EITHER OVERFLOW OR UNDERFLOW. A CALL
                              ;TO IT IS MADE THUS:
                              ;
                              ;                ACARG:  .WORD   X,X,X,X          ;AC OPERAND
                              ;                FSRCARG:.WORD   X,X,X,X          ;FSRC OPERAND
                              ;                RES:    .WORD   X,X,X,X          ;EXPECTED RESULT
                              ;                ERRES:  .WORD   X,X,X,X          ;ERROR RESULT
                              ;                FPSB:   .WORD   X                ;FPS BEFORE EXECUTION
                              ;                FPSA:   .WORD   X                ;FPS AFTER EXECUTION
                              ;                FEC:    .WORD   X                ;EXPECTED FEC
                              ;                FLAG:   .WORD   X                ;0/-1,OVER/UNDER FLOW FLAG
                              ;                ERR1:   ERROR   X                ;TRAP ERROR.
                              ;                BR      CONT
                              ;                ERR2:   ERROR   X                ;DATA, RESULT ERROR
                              ;                CONT:                            ;RETURN ADDRESS
                              ;
                              ;THE OPERANDS ARE SET UP (USING AC0 AS THE ACCUMULATOR). THEN
                              ;THE MULD INSTRUCTION IS EXECUTED. IF THE TRAP OCCURS THEN THE
                              ;RESULT IS CHECKED AGAINST RES. IF THE RESULT IS CORRECT THEN THE FPS IS
                              ;COMPARED WITH FPSA IF THIS TOO IS CORRECT OVUNDT RETURNS CONTROL
```

```
                                       ;TO THE CALLING ROUTINE AT CONT. IF THE FPS IS BAD OVUNDT
                                       ;REPORTS THIS FAILURE AND THEN RETURNS TO CONT. THE FEC IS TREATED
                                       ;IN THE SAME WAY. IF THE RESULT OF THE
                                       ;MULF IS INCORRECT, THE INCORRECT RESULT IS COMPARED WITH THE
                                       ;ANTICIPATED FAILING DATA PATTERN, ERRES. IF THE FAILURE IN
                                       ;THE RESULT WAS ANTICIPATED CORRECTLY TO BE ERRES THEN OVUNDT
                                       ;WILL TRANSFER CONTROL TO THE ERROR CALL AT ERR2. OTHERWISE THE
                                       ;RESULT WAS INCORRECT BUT WAS NOT ANTICIPATED AND OVUNDT WILL
                                       ;REPORT THE FAILURE AFTER WHICH CONTROL WILL BE PASSED TO CONT.
                                       ;IF NO TRAP OCCURS CONTROL IS PASSED TO ERR1.
                                       ;NOTE THAT OVUNDNT USES THE FLAG
                                       ;TO TELL WHETHER OR NOT THESE PARTICULAR OPERANDS WILL RESULT IN
                                       ;UNDERFLOW (FLAG=-1) OR OVERFLOW (FLAG=0).

074342  012601                         OVUNDT: MOV     (SP)+,R1        ;GET A POINTER TO THE ARGUMENTS.
074344  012700  000200                         MOV     #200,R0         ;SET FD MODE.
074350  170100                                 LDFPS   R0

074352  010100                                 MOV     R1,R0           ;LOAD AC0, OPERAND.
074354  172410                                 LDD     (R0),AC0
074356  016100  000030                         MOV     30(R1),R0       ;LOAD THE FPS.
074362  170100                                 LDFPS   R0
074364  012737  074406  000244                 MOV     #50$,@#FPVECT   ;SET UP THE FP TRAP VECTOR IN CASE
                                                                       ;OF ERROR.
074372  010100                                 MOV     R1,R0           ;COMPUTE THE ADDRESS OF FSRC.
074374  062700  000010                         ADD     #10,R0

074400  171010                         1$:     MULD    (R0),AC0        ;TEST INSTRUCTION. SHOULD CAUSE TRAP.
074402  170000                         2$:     CFCC
074404  104000                                 EMT                     ;
074406  011602                         50$:    MOV     (SP),R2         ;TRAP TO HERE AND SEE IF THE PC OF THE
074410  020227  074402                         CMP     R2,#2$          ;TRAP WAS THAT OF THE MULF INSTRUCTION.
074414  001401                                 BEQ     51$             ;BRANCH IF YES.
074416  104000                                 EMT                     ;
074420  022626                         51$:    CMP     (SP)+,(SP)+     ;RESET THE STACK
074422  170204                                 STFPS   R4              ;GET FPS.
074424  170305                                 STST    R5              ;GET FEC.
074426  012700  000200                         MOV     #200,R0         ;SET FD MODE.
074432  170100                                 LDFPS   R0
074434  012700  074514                         MOV     #OVDTT,R0       ;GET THE RESULT.
074440  174010                                 STD     AC0,(R0)
074442  012700  074514                         MOV     #OVDTT,R0       ;CHECK THE RESULT.
074446  010102                                 MOV     R1,R2
074450  062702  000020                         ADD     #20,R2
074454  012703  000004                         MOV     #4,R3
074460  022022                         3$:     CMP     (R0)+,(R2)+
074462  001401                                 BEQ     5$
074464  104000                                 EMT                     ;
074466  077304                         5$:     SOB     R3,3$
074470  026104  000032                         CMP     32(R1),R4       ;WAS FPS CORRECT?
074474  001401                                 BEQ     6$
074476  104000                                 EMT                     ;
074500  026105  000034                 6$:     CMP     34(R1),R5       ;IS FEC CORRECT?
074504  001401                                 BEQ     4$
074506  104000                                 EMT                     ;
074510  000161  000040                 4$:     JMP     40(R1)          ;RETURN, TEST COMPLETED.
```

```
074514  000000  000000  000000  OVDTT:  .WORD   0,0,0,0
074522  000000

074524                          LLLDONE:
074524  004767  023744                  JSR     PC,.RSET        ;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?).

                                ;*************************************************************************
                                ;TEST 473       MODF TEST
                                ;*************************************************************************
074530                          TS473:

                                ;MODF WITH (FSRC=AC=0)
074530  004737  075254          GGG1:   JSR     PC,@#MODFSUB
074534  000000  000000          1$:     .WORD   0,0             ;AC
074540  000000  000000          2$:     .WORD   0,0             ;FSRC
074544  000000  000000          3$:     .WORD   0,0             ;FRACTIONAL RES.
074550  000000  000000          4$:     .WORD   0,0             ;INTEGER RES.
074554  000013                  7$:     13                      ;FPS BEFORE EXECUTION.
074556  000004                          4                       ;FPS AFTER EXECUTION.

                                ;MODF TEST, WITH (FSRC=0)
074560  004737  075254          GGG2:   JSR     PC,@#MODFSUB
074564  123456  076543          1$:     .WORD   123456,76543    ;AC
074570  000000  000000          2$:     .WORD   0,0             ;FSRC
074574  000000  000000          3$:     .WORD   0,0             ;FRACTIONAL RES.
074600  000000  000000          4$:     .WORD   0,0             ;INTEGER RESULT.
074604  000000                  7$:     0                       ;FPS BEFORE EXECUTION.
074606  000004                          4                       ;FPS AFTER EXECUTION.

                                ;MODF TEST WITH (AC=0)
074610  004737  075254          GGG3:   JSR     PC,@#MODFSUB
074614  000000  000000          1$:     .WORD   0,0             ;AC
074620  076543  021234          2$:     .WORD   76543,21234     ;FSRC
074624  000000  000000          3$:     .WORD   0,0             ;FRACTIONAL RES.
074630  000000  000000          4$:     .WORD   0,0             ;INTEGER RES.
074634  000003                  7$:     3                       ;FPS BEFORE EXECUTION.
074636  000004                          4                       ;FPS AFTER EXECUTION.

                                ;MODF TEST WITH EXPONENT OF THE RESULT = 25
074640  004737  075254          GGG4:   JSR     PC,@#MODFSUB
074644  046252  125252          1$:     .WORD   46252,125252    ;AC
074650  040300  000000          2$:     .WORD   40300,0         ;FSRC
074654  000000  000000          3$:     .WORD   0,0             ;FRACTIONAL RES.
074660  046377  177777          4$:     .WORD   46377,-1        ;INTEGER RES.
074664  000013                  7$:     13                      ;FPS BEFORE EXECUTION.
074666  000004                          4                       ;FPS AFTER EXECUTION.
```

```
                                        ;MODF TEST WITH EXPONENT OF THE RESULT = 127
074670  004737  075254                  GGG5:   JSR     PC,@#MODFSUB
074674  077652  125252                  1$:     .WORD   77652,125252    ;AC
074700  040300  000000                  2$:     .WORD   40300,0         ;FSRC
074704  000000  000000                  3$:     .WORD   0,0             ;FRACTIONAL RES.
074710  077777  177777                  4$:     .WORD   77777,-1        ;INTEGER RES.
074714  000000                          7$:     0                       ;FPS BEFORE EXECUTION.
074716  000004                                  4                       ;FPS AFTER EXECUTION.

                                        ;MODF TEST WITH EXPONENT OF RESULT = 25
074720  004737  075254                  GGG6:   JSR     PC,@#MODFSUB
074724  046200  000001                  1$:     .WORD   46200,1         ;AC
074730  040340  000000                  2$:     .WORD   40340,0         ;FSRC
074734  000000  000000                  3$:     .WORD   0,0             ;FRACTIONAL RES.
074740  046340  000001                  4$:     .WORD   46340,1         ;INTEGER RES.
074744  000013                          7$:     13                      ;FPS BEFORE EXECUTION.
074746  000004                                  4                       ;FPS AFTER EXECUTION.

                                        ;MODF   TEST WITH EXPONENT OF THE RESULT = 24
074750  004737  075254                  GGG7:   JSR     PC,@#MODFSUB
074754  046000  000001                  1$:     .WORD   46000,1         ;AC
074760  040340  000000                  2$:     .WORD   40340,0         ;FSRC
074764  040100  000000                  3$:     .WORD   40100,0         ;FRACTIONAL RES.
074770  046140  000001                  4$:     .WORD   46140,1         ;INTEGER RESULT.
074774  000000                          7$:     0                       ;FPS BEFORE EXECUTION.
074776  000000                                  0                       ;FPS AFTER EXECUTION.

                                        ;MODF TEST WITH EXPONENT OF THE RESULT = 10
075000  004737  075254                  GGG8:   JSR     PC,@#MODFSUB
075004  042577  177777                  1$:     .WORD   42577,-1        ;AC
075010  040200  000000                  2$:     .WORD   40200,0         ;FSRC
075014  040177  176000                  3$:     .WORD   40177,176000    ;FRACTIONAL RES.
075020  042577  140000                  4$:     .WORD   42577,140000    ;INTEGER RES.
075024  000000                          7$:     0                       ;FPS BEFORE EXECUTION.
075026  000000                                  0                       ;FPS AFTER EXECUTION.

                                        ;MODF TEST WITH THE EXPONENT OF THE RESULT = 10
075030  004737  075254                  GGG9:   JSR     PC,@#MODFSUB
075034  042577  140001                  1$:     .WORD   42577,140001    ;AC
075040  040200  000000                  2$:     .WORD   40200,0         ;FSRC
075044  034600  000000                  3$:     .WORD   34600,0         ;FRACTIONAL RES.
075050  042577  140000                  4$:     .WORD   42577,140000    ;INTEGER RES.
075054  000000                          7$:     0                       ;FPS BEFORE EXECUTION.
075056  000000                                  0                       ;FPS AFTER EXECUTION.

                                        ;MODF TEST WITH EXPONENT OF THE RESULT = 9
075060  004737  075254                  GGG10:  JSR     PC,@#MODFSUB
075064  042377  100000                  1$:     .WORD   42377,100000    ;AC
075070  040200  000000                  2$:     .WORD   40200,0         ;FSRC
075074  000000  000000                  3$:     .WORD   0,0             ;FRACTIONAL RES.
075100  042377  100000                  4$:     .WORD   42377,100000    ;INTEGER RES.
075104  000013                          7$:     13                      ;FPS BEFORE EXECUTION.
075106  000004                                  4                       ;FPS AFTER EXECUTION.

                                        ;MODF TEST WITH EXPONENT OF THE RESULT = 0
075110  004737  075254                  GGG11:  JSR     PC,@#MODFSUB
```

```
075114  040177  177777          1$:     .WORD   40177,-1        ;AC
075120  040200  000000          2$:     .WORD   40200,0         ;FSRC
075124  040177  177777          3$:     .WORD   40177,-1        ;FRACTIONAL RES.
075130  000000  000000          4$:     .WORD   0,0             ;INTEGER RES.
075134  000017                  7$:     17                      ;FPS BEFORE EXECUTION.
075136  000000                          0                       ;FPS AFTER EXECUTION.


                                ;MODF TEST WITH EXPONENT OF THE RESULT = -15
075140  004737  075254          GGG12:  JSR     PC,@#MODFSUB
075144  034377  177777          1$:     .WORD   34377,-1        ;AC
075150  040200  000000          2$:     .WORD   40200,0         ;FSRC
075154  034377  177777          3$:     .WORD   34377,-1        ;FRACTIONAL RES.
075160  000000  000000          4$:     .WORD   0,0             ;INTEGER RES.
075164  000000                  7$:     0                       ;FPS BEFORE EXECUTION.
075166  000000                          0                       ;FPS AFTER EXECUTION.


                                ;MODF TEST WITH EXPONENT OF RESULT = -64, IN ROUND MODE
075170  004737  075254          GGG13:  JSR     PC,@#MODFSUB
075174  020000  000001          1$:     .WORD   20000,1         ;AC
075200  040300  000000          2$:     .WORD   40300,0         ;FSRC
075204  020100  000002          3$:     .WORD   20100,2         ;FRACTIONAL RES.
075210  000000  000000          4$:     .WORD   0,0             ;INTEGER RES.
075214  000000                  7$:     0                       ;FPS BEFORE EXECUTION.
075216  000000                          0                       ;FPS AFTER EXECUTION.


                                ;MODF TEST WITH EXPONENT OF RESULT = 11
075220  004737  075254          GGG14:  JSR     PC,@#MODFSUB
075224  142777  170000          1$:     .WORD   142777,170000   ;AC
075230  040200  000000          2$:     .WORD   40200,0         ;FSRC
075234  140000  000000          3$:     .WORD   140000,0        ;FRACTIONAL RES.
075240  142777  160000          4$:     .WORD   142777,160000   ;INTEGER RES.
075244  000007                  7$:     7                       ;FPS BEFORE EXECUTION.
075246  000010                          10                      ;FPS AFTER EXECUTION.
075250  000167  000204          9$:     JMP     GGGDONE         ;GO TO NEXT TEST.


                                ;THIS SUBROUTINE, MODFSUB, IS CALLED TO SETUP THE
                                ;OPERANDS, EXECUTE THE MODF INSTRUCTION AND CHECK THE RESULTS.
                                ;IT IS CALLED THUS:
                                ;
                                ;               ACARG:  .WORD   X,X             ;AC OPERAND
                                ;               FSRCARG:.WORD   X,X             ;FSRC OPERAND
                                ;               FRES:   .WORD   X,X             ;FRACTIONAL RESULT
                                ;               INTRES: .WORD   X,X             ;INTEGER RESULT
                                ;               ERFRES: .WORD   X,X             ;ERROR FRACTION RESULT
                                ;               ERINTRES:.WORD  X,X             ;ERROR INTEGER RESULT
                                ;               FPSB:   .WORD   X               ;FPS BEFORE EXECUTION
                                ;               FPSA:   .WORD   X               ;FPS AFTER EXECUTION
                                ;               ERR1:   ERROR   X               ;FRACTION ERROR
                                ;                       BR      CONT
                                ;               ERR2:   ERROR   X               ;INTEGER ERROR
                                ;               CONT:                           ;RETURN ADDRESS
                                ;
                                ;THE OPERANDS ARE SET UP (USING AC0 FOR THE AC ARGUMENT). THE MODF
                                ;INSTRUCTION IS EXECUTED. THEN THE RESULTS ARE RETRIEVED.
                                ;THE FRACTION PART OF THE RESULT IS COMPARED WITH FRES. IF THIS IS CORRECT
```

```
                                     ;THEN THE INTEGER PART IS COMPARED WITH INTRES. IF BOTH OF THESE ARE CORRECT
                                     ;THEN THE FPS IS COMPARED WITH FPSA. AFTER EXECUTION IF NO ERRORS OCCURRED
                                     ;THEN MODFSUB WILL RETURN TO CONT. IF THE FPS WAS INCORRECT
                                     ;IT IS REPORTED HERE. IF THE FRACTION IS INCORRECT IT IS COMPARED WITH
                                     ;THE ANTICIPATED BAD FRACTION, ERFRES. IF THIS DOESN'T MATCH
                                     ;THE TRUE RESULT THEN THE ERROR IS REPORTED HERE. IF THE ANTICIPATED
                                     ;FAILURE MATCHES THE TRUE RESULT THEN  MODFSUB PASSES CONTROL TO THE
                                     ;ERROR CALL AT ERR1. LIKEWISE IF THE INTEGER PART OF THE RESULT IS
                                     ;NOT CORRECT THEN IT IS COMPARED WITH THE ANTICIPATED INTEGER
                                     ;FAILURE. IF THIS DOEN'T MATCH THEN THE ERROR IS REPORTED HERE.
                                     ;IF A MATCH IS MADE HOWEVER, MODFSUB WILL RETURN CONTROL TO THE ERROR
                                     ;CALL AT ERR2.

075254  012601                       MODFSUB:        MOV     (SP)+,R1        ;GET A POINTER TO THE ARGUMENTS
075256  012700  000200                       MOV     #200,R0         ;SET FD MODE.
075262  170100                               LDFPS   R0
075264  010100                               MOV     R1,R0           ;SET UP AC0
075266  172410                               LDD     (R0),AC0
075270  012700  075450                       MOV     #MODP1,R0       ;PUT A BACKROUND PATTERN INTO AC1.
075274  172510                               LDD     (R0),AC1
075276  016100  000020                       MOV     20(R1),R0       ;SET UP THE FFS.
075302  170100                               LDFPS   R0
075304  010100                               MOV     R1,R0           ;COMPUTE THE ADDRESS OF THE FSRC.
075306  062700  000004                       ADD     #4,R0

075312  171410                       1$:     MODF    (R0),AC0        ;EXECUTE THE TEST INSTRUCTION.

075314  170204                               STFPS   R4              ;GET THE FPS.
075316  012700  000200                       MOV     #200,R0         ;SET FD MODE.
075322  170100                               LDFPS   R0
075324  012700  075430                       MOV     #MODFT0,R0      ;GET THE FRACTIONAL RESULT.
075330  174010                               STD     AC0,(R0)
075332  012700  075440                       MOV     #MODFT1,R0      ;GET THE INTEGER RESULT.
075336  174110                               STD     AC1,(R0)
075340  012702  075430                       MOV     #MODFT0,R2      ;CHECK THE FRACTIONAL RESULT.
075344  026112  000010                       CMP     10(R1),(R2)
075350  001401                               BEQ     2$
075352  104000                               EMT                     ;
075354  026162  000012  000002       2$:     CMP     12(R1),2(R2)
075362  001401                               BEQ     3$
075364  104000                               EMT                     ;
075366  012702  075440               3$:     MOV     #MODFT1,R2      ;CHECK THE INTEGER RESULT.
075372  026112  000014                       CMP     14(R1),(R2)
075376  001401                               BEQ     4$
075400  104000                               EMT                     ;
075402  026162  000016  000002       4$:     CMP     16(R1),2(R2)
075410  001401                               BEQ     5$
075412  104000                               EMT                     ;
075414  026104  000022               5$:     CMP     22(R1),R4       ;CHECK THE FPS.
075420  001401                               BEQ     9$
075422  104000                               EMT                     ;
075424  000161  000024               9$:     JMP     24(R1)          ;RETURN.

075430  000000  000000  000000       MODFT0: .WORD   0,0,0,0
075436  000000

```

```
075440  000000  000000  000000  MODFT1: .WORD   0,0,0,0
075446  000000

075450  177777  177777  177777  MODP1:  .WORD   -1,-1,-1,-1
075456  177777

075460                          GGGDONE:
075460  004767  023010                  JSR     PC,.RSET        ;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?).





                                ;*************************************************************************
                                ;TEST 474       MODD TEST
                                ;*************************************************************************
075464                          TS474:

                                ;MODD WITH (FSRC=AC=0)
075464  004737  076500          HHH1:   JSR     PC,@#MODDSUB
075470  000000  000000  000000  1$:     .WORD   0,0,0,0         ;AC
075476  000000
075500  000000  000000  000000  2$:     .WORD   0,0,0,0         ;FSRC
075506  000000
075510  000000  000000  000000  3$:     .WORD   0,0,0,0         ;FRACTIONAL RES.
075516  000000
075520  000000  000000  000000  4$:     .WORD   0,0,0,0         ;INTEGER RES.
075526  000000
075530  000200                  7$:     200                     ;FPS BEFORE EXECUTION.
075532  000204                          204                     ;FPS AFTER EXECUTION.

                                ;MODD TEST WITH FSRC=0
075534  004737  076500          HHH2:   JSR     PC,@#MODDSUB
075540  012345  067012          1$:     .WORD   012345,67012    ;AC
075544  034567  012345                  .WORD   34567,012345
075550  000000  000000  000000  2$:     .WORD   0,0,0,0         ;FSRC
075556  000000
075560  000000  000000  000000  3$:     .WORD   0,0,0,0         ;FRACTIONAL RES.
075566  000000
075570  000000  000000  000000  4$:     .WORD   0,0,0,0         ;INTEGER RES.
075576  000000
075600  000213                  7$:     213                     ;FPS BEFORE EXECUTION.
075602  000204                          204                     ;FPS AFTER EXECUTION.

                                ;MODD TEST WITH (AC=0)
075604  004737  076500          HHH3:   JSR     PC,@#MODDSUB
075610  000000  000000  000000  1$:     .WORD   0,0,0,0         ;AC
075616  000000
075620  072727  127272          2$:     .WORD   72727,127272    ;FSRC
075624  072727  127272                  .WORD   72727,127272
075630  000000  000000  000000  3$:     .WORD   0,0,0,0         ;FRACTIONAL RES.
075636  000000
075640  000000  000000  000000  4$:     .WORD   0,0,0,0         ;INTEGER RES.
```

```
075646  000000
075650  000213                          7$:     213                             ;FPS BEFORE EXECUTION.
075652  000204                                  204                             ;FPS AFTER EXECUTION.

                                        ;MODD TEST WITH EXPONENT OF THE RESULT = 57
075654  004737  076500                  HHH4:   JSR     PC,@#MODDSUB
075660  056252  125252                  1$:     .WORD   56252,125252            ;AC
075664  125252  125250                          .WORD   125252,125250
075670  040300  000000  000000          2$:     .WORD   40300,0,0,0             ;FSRC
075676  000000
075700  000000  000000  000000          3$:     .WORD   0,0,0,0                 ;FRACTIONAL RES.
075706  000000
075710  056377  177777  177777          4$:     .WORD   56377,-1,-1,-4          ;INTEGER RES.
075716  177774
075720  000213                          7$:     213                             ;FPS BEFORE EXECUTION.
075722  000204                                  204                             ;FPS AFTER EXECUTION.

                                        ;MODD TEST WITH EXPONENT OF THE RESULT = 79
075724  004737  076500                  HHH5:   JSR     PC,@#MODDSUB
075730  140240  000000  000000          1$:     .WORD   140240,0,0,0            ;AC
075736  000000
075740  063714  146314                  2$:     .WORD   63714,146314            ;FSRC
075744  133572  167737                          .WORD   133572,167737
075750  000000  000000  000000          3$:     .WORD   0,0,0,0                 ;FRACTIONAL RES.
075756  000000
075760  163777  177777                  4$:     .WORD   163777,-1               ;INTEGER RES.
075764  162531  125726                          .WORD   162531,125726
075770  000210                          7$:     210                             ;FPS BEFORE EXECUTION.
075772  000204                                  204                             ;FPS AFTER EXECUTION.

                                        ;MODD TEST WITH EXPONENT OF THE RESULT = 57
075774  004737  076500                  HHH6:   JSR     PC,@#MODDSUB
076000  056200  000000  000000          1$:     .WORD   56200,0,0,1             ;AC
076006  000001
076010  040340  000000  000000          2$:     .WORD   40340,0,0,0             ;FSRC
076016  000000
076020  000000  000000  000000          3$:     .WORD   0,0,0,0                 ;FRACTIONAL RES.
076026  000000
076030  056340  000000  000000          4$:     .WORD   56340,0,0,1             ;INTEGER RES.
076036  000001
076040  000213                          7$:     213                             ;FPS BEFORE EXECUTION.
076042  000204                                  204                             ;FPS AFTER EXECUTION.

                                        ;MODD TEST WITH EXPONENT OF THE RESULT = 56
076044  004737  076500                  HHH7:   JSR     PC,@#MODDSUB
076050  056000  000000  000000          1$:     .WORD   56000,0,0,1             ;AC
076056  000001
076060  040340  000000  000000          2$:     .WORD   40340,0,0,0             ;FSRC
076066  000000
076070  040100  000000  000000          3$:     .WORD   40100,0,0,0             ;FRACTIONAL RES.
076076  000000
076100  056140  000000  000000          4$:     .WORD   56140,0,0,1             ;INTEGER RES.
076106  000001
076110  000213                          7$:     213                             ;FPS BEFORE EXECUTION.
076112  000200                                  200                             ;FPS AFTER EXECUTION.
```

```
                                    ;MODD TEST WITH EXPONENT OF THE RESULT = 36
076114  004737  076500              HHH8:   JSR     PC,@#MODDSUB
076120  051177  177777  177777      1$:     .WORD   51177,-1,-1,-1  ;AC
076126  177777
076130  040200  000000  000000      2$:     .WORD   40200,0,0,0     ;FSRC
076136  000000
076140  040177  177760  000000      3$:     .WORD   40177,-20,0,0   ;FRACTIONAL RES.
076146  000000
076150  051177  177777  177760      4$:     .WORD   51177,-1,-20,0  ;INTEGER RES.
076156  000000
076160  000217                      7$:     217                     ;FPS BEFORE EXECUTION.
076162  000200                              200                     ;FPS AFTER EXECUTION.

                                    ;MODD TEST WITH EXPONENT OF THE RESULT = 30
076164  004737  076500              HHH9:   JSR     PC,@#MODDSUB
076170  040200  000000  000000      1$:     .WORD   40200,0,0,0     ;AC
076176  000000
076200  047577  177777              2$:     .WORD   47577,-1        ;FSRC
076204  176000  000001                      .WORD   176000,1
076210  031600  000000  000000      3$:     .WORD   31600,0,0,0     ;FRACTIONAL RES.
076216  000000
076220  047577  177777              4$:     .WORD   47577,-1        ;INTEGER RES.
076224  176000  000000                      .WORD   176000,0
076230  000200                      7$:     200                     ;FPS BEFORE EXECUTION.
076232  000200                              200                     ;FPS AFTER EXECUTION.

                                    ;MODD TEST WITH EXPONENT OF THE RESULT = 31
076234  004737  076500              HHH10:  JSR     PC,@#MODDSUB
076240  047777  177777              1$:     .WORD   47777,-1        ;AC
076244  177000  000000                      .WORD   177000,0
076250  040200  000000  000000      2$:     .WORD   40200,0,0,0     ;FSRC
076256  000000
076260  000000  000000  000000      3$:     .WORD   0,0,0,0         ;FRACTIONAL RES.
076266  000000
076270  047777  177777              4$:     .WORD   47777,-1        ;INTEGER RES.
076274  177000  000000                      .WORD   177000,0
076300  000213                      7$:     213                     ;FPS BEFORE EXECUTION.
076302  000204                              204                     ;FPS AFTER EXECUTION.

                                    ;MODD TEST WITH EXPONENT OF THE RESULT = 0
076304  004737  076500              HHH11:  JSR     PC,@#MODDSUB
076310  040200  000000  000000      1$:     .WORD   40200,0,0,0     ;AC
076316  000000
076320  040177  072727              2$:     .WORD   40177,72727     ;FSRC
076324  127272  072727                      .WORD   127272,72727
076330  040177  072727              3$:     .WORD   40177,72727     ;FRACTIONAL RES.
076334  127272  072727                      .WORD   127272,72727
076340  000000  000000  000000      4$:     .WORD   0,0,0,0         ;INTEGER RES.
076346  000000
076350  000200                      7$:     200                     ;FPS BEFORE EXECUTION.
076352  000200                              200                     ;FPS AFTER EXECUTION.

                                    ;MODD TEST WITH EXPONENT OF THE RESULT = -115
076354  004737  076500              HHH12:  JSR     PC,@#MODDSUB
076360  003377  177777              1$:     .WORD   3377,-1         ;AC
076364  177777  052525                      .WORD   -1,52525
```

```
076370  040200  000000  000000  2$:     .WORD   40200,0,0,0         ;FSRC
076376  000000
076400  003377  177777          3$:     .WORD   3377,-1             ;FRACTIONAL RES.
076404  177777  052525                  .WORD   -1,52525
076410  000000  000000  000000  4$:     .WORD   0,0,0,0             ;INTEGER RES.
076416  000000
076420  000200                  7$:     200                         ;FPS BEFORE EXECUTION.
076422  000200                          200                         ;FPS AFTER EXECUTION.

                                ;MODD TEST WITH EXPONENT OF THE RESULT = -63, IN ROUND MODE.
076424  004737  076500          HHH13:  JSR     PC,@#MODDSUB
076430  040300  000000  000000  1$:     .WORD   40300,0,0,0         ;AC
076436  000000
076440  020200  000000  000000  2$:     .WORD   20200,0,0,1         ;FSRC
076446  000001
076450  020300  000000  000000  3$:     .WORD   20300,0,0,2         ;FRACTIONAL RES.
076456  000002
076460  000000  000000  000000  4$:     .WORD   0,0,0,0             ;INTEGER RES.
076466  000000
076470  000200                  7$:     200                         ;FPS BEFORE EXECUTION.
076472  000200                          200                         ;FPS AFTER EXECUTION.
076474  000137  076674          9$:     JMP     @#HHHDONE           ;GO TO THE NEXT TEST.

                                ;THIS SUBROUTINE, MODDSUB, IS CALLED TO SETUP THE
                                ;OPERANDS, EXECUTE THE MODD INSTRUCTION AND CHECK THE RESULTS.
                                ;IT IS CALLED THUS:
                                ;
                                ;               ACARG:  .WORD   X,X,X,X         ;AC OPERAND
                                ;               FSRCARG:.WORD   X,X,X,X         ;FSRC OPERAND
                                ;               FRES:   .WORD   X,X,X,X         ;FRACTIONAL RESULT
                                ;               INTRES: .WORD   X,X,X,X         ;INTEGER RESULT
                                ;               ERFRES: .WORD   X,X,X,X         ;ERROR FRACTION RESULT
                                ;               ERINTRES:.WORD  X,X,X,X         ;ERROR INTEGER RESULT
                                ;               FPSB:   .WORD   X               ;FPS BEFORE EXECUTION
                                ;               FPSA:   .WORD   X               ;FPS AFTER EXECUTION
                                ;               ERR1:   ERROR   X               ;FRACTION ERROR
                                ;                       BR      CONT
                                ;               ERR2:   ERROR   X               ;INTEGER ERROR
                                ;               CONT:                           ;RETURN ADDRESS
                                ;
                                ;THE OPERANDS ARE SET UP (USING AC0 FOR THE AC ARGUMENT). THE MODD
                                ;INSTRUCTION IS EXECUTED. THEN THE RESULTS ARE RETRIEVED.
                                ;THE FRACTION PART OF THE RESULT IS COMPARED WITH FRES. IF THIS IS CORRECT
                                ;THEN THE INTEGER PART IS COMPARED WITH INTRES. IF BOTH OF THESE ARE CORRECT
                                ;THEN THE FPS IS COMPARED WITH FPSA. AFTER EXECUTION IF NO ERRORS OCCURRED
                                ;THEN MODDSUB WILL RETURN TO CONT. IF THE FPS WAS INCORRECT
                                ;IT IS REPORTED HERE. IF THE FRACTION IS INCORRECT IT IS COMPARED WITH
                                ;THE ANTICIPATED BAD FRACTION, ERFRES. IF THIS DOESN'T MATCH
                                ;THE TRUE RESULT THEN THE ERROR IS REPORTED HERE. IF THE ANTICIPATED
                                ;FAILURE MATCHES THE TRUE RESULT THEN  MODDSUB PASSES CONTROL TO THE
                                ;ERROR CALL AT ERR1. LIKEWISE IF THE INTEGER PART OF THE RESULT IS
                                ;NOT CORRECT THEN IT IS COMPARED WITH THE ANTICIPATED INTEGER
                                ;FAILURE. IF THIS DOEN'T MATCH THEN THE ERROR IS REPORTED HERE.
                                ;IF A MATCH IS MADE HOWEVER, MODDSUB WILL RETURN CONTROL TO THE ERROR
                                ;CALL AT ERR2.
```

```
076500  012601                          MODDSUB:        MOV     (SP)+,R1        ;GET A POINTER TO THE ARGUMENTS
076502  012700  000200                          MOV     #200,R0         ;SET FD MODE.
076506  170100                                  LDFPS   R0
076510  010100                                  MOV     R1,R0           ;SET UP AC0
076512  172410                                  LDD     (R0),AC0
076514  012700  075450                          MOV     #MODP1,R0       ;PUT A BACKROUND PATTERN INTO AC1.
076520  172510                                  LDD     (R0),AC1
076522  016100  000040                          MOV     40(R1),R0       ;SET UP THE FPS.
076526  170100                                  LDFPS   R0
076530  010100                                  MOV     R1,R0           ;COMPUTE THE ADDRESS OF THE FSRC.
076532  062700  000010                          ADD     #10,R0

076536  171410                          1$:     MODD    (R0),AC0        ;EXECUTE THE TEST INSTRUCTION.

076540  170204                                  STFPS   R4              ;GET THE FPS.
076542  012700  000200                          MOV     #200,R0         ;SET FD MODE.
076546  170100                                  LDFPS   R0
076550  012700  076654                          MOV     #MODDT0,R0      ;GET THE FRACTIONAL PESULT.
076554  174010                                  STD     AC0,(R0)
076556  012700  076664                          MOV     #MODDT1,R0      ;GET THE INTEGER RESULT.
076562  174110                                  STD     AC1,(R0)
076564  012702  076654                          MOV     #MODDT0,R2      ;CHECK THE FRACTIONAL RESULT.
076570  010103                                  MOV     R1,R3
076572  062703  000020                          ADD     #20,R3
076576  012705  000004                          MOV     #4,R5
076602  022223                          2$:     CMP     (R2)+,(R3)+
076604  001401                                  BEQ     4$
076606  104000                                  EMT                     ;
076610  077504                          4$:     SOB     R5,2$
076612  012702  076664                          MOV     #MODDT1,R2      ;CHECK THE INTEGER RESULT.
076616  010103                                  MOV     R1,R3
076620  062703  000030                          ADD     #30,R3
076624  012705  000004                          MOV     #4,R5
076630  022223                          3$:     CMP     (R2)+,(R3)+
076632  001401                                  BEQ     5$
076634  104000                                  EMT                     ;
076636  077504                          5$:     SOB     R5,3$
076640  026104  000042                          CMP     42(R1),R4       ;CHECK THE FPS.
076644  001401                                  BEQ     9$
076646  104000                                  EMT                     ;
076650  000161  000044                  9$:     JMP     44(R1)          ;RETURN.

076654  000000  000000  000000  MODDT0: .WORD   0,0,0,0
076662  000000

076664  000000  000000  000000  MODDT1: .WORD   0,0,0,0
076672  000000

076674                                  HHHDONE:
076674  004767  021574                          JSR     PC,.RSET        ;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?).
```

```



                           ;*************************************************************************
                           ;TEST 475        UNDER\OVER FLOW, USING MODF WITH TRAPS DISABLED, TEST
                           ;*************************************************************************
076700                     TS475:

                           ;UNDERFLOW TEST, WITH EXPONENT OF THE RESULT = -129, FIU = 1, FID = 1
076700 004767 000214       MMM1:   JSR     PC,MODFOV
076704 020123 045676       1$:     .WORD   20123,45676     ;AC
076710 020200 000000       2$:     .WORD   20200,0         ;FSRC
076714 000123 045676       3$:     .WORD   123,45676       ;FRACTIONAL RES.
076720 000000 000000       4$:     .WORD   0,0             ;INTEGER RES.
076724 042000              7$:     42000                   ;FPS BEFORE EXECUTION.
076726 142004                      142004                  ;FPS AFTER EXECUTION.
076730 000012                      12                      ;FEC
076732 104000                      EMT                     ;

                           ;UNDERFLOW EXP OF RESULT = -193, FIU = 0, FID = 1
076734 004737 077120       MMM2:   JSR     PC,@#MODFOV
076740 010200 000000       1$:     .WORD   10200,0         ;AC
076744 010000 000000       2$:     .WORD   10000,0         ;FSRC
076750 000000 000000       3$:     .WORD   0,0             ;FRACTIONAL RES.
076754 000000 000000       4$:     .WORD   0,0             ;INTEGER RES.
076760 005013              7$:     5013                    ;FPS BEFORE EXECUTION.
076762 005004                      5004                    ;FPS AFTER EXECUTION.
076764 000012                      12                      ;FEC
076766 000240                      NOP

                           ;OVERFLOW TEST WITH EXPONENT OF THE RESULT = 128, FIV = 1, FID = 1
076770 004737 077120       MMM3:   JSR     PC,@#MODFOV
076774 060052 125252       1$:     .WORD   60052,125252    ;AC
077000 060200 000000       2$:     .WORD   60200,0         ;FSRC
077004 000000 000000       3$:     .WORD   0,0             ;FRACTIONAL RES.
077010 000052 125252       4$:     .WORD   52,125252       ;INTEGER RES.
077014 041000              7$:     41000                   ;FPS BEFORE EXECUTION.
077016 141006                      141006                  ;FPS AFTER EXECUTION.
077020 000010                      10                      ;FEC
077022                     8$:
077022 104000                      EMT                     ;
                           ;OVERFLOW TEST WITH EXPONENT OF THE RESULT = 130, FIV = 0, FID = 1
077024 004737 077120       MMM4:   JSR     PC,@#MODFOV
077030 060345 067654       1$:     .WORD   60345,67654     ;AC
077034 060200 000000       2$:     .WORD   60200,0         ;FSRC
077040 000000 000000       3$:     .WORD   0,0             ;FRACTIONAL RES.
077044 000000 000000       4$:     .WORD   0,0             ;INTEGER RES.
077050 006011              7$:     6011                    ;FPS BEFORE EXECUTION.
077052 006006                      6006                    ;FPS AFTER EXECUTION.
077054 000010                      10                      ;FEC
077056 000240              8$:     NOP
                           ;OVERFLOW TEST WITH EXPONENT OF THE RESULT = 128, RESULT NEGATIVE
                           ;AND FIV = 1, FID = 1
077060 004737 077120       MMM5:   JSR     PC,@#MODFOV
077064 160252 125252       1$:     .WORD   160252,125252   ;AC
077070 060000 000000       2$:     .WORD   60000,0         ;FSRC
077074 000000 000000       3$:     .WORD   0,0             ;FRACTIONAL RES.
```

```
077100  100052  125252     4$:     .WORD   100052,125252      ;INTEGER RES.
077104  041000             7$:     41000                      ;FPS BEFORE EXECUTION.
077106  141006                     141006                     ;FPS AFTER EXECUTION.
077110  000010                     10                         ;FEC
077112                     8$:
077112  104000                     EMT                        ;
077114  000137  077334     9$:     JMP     @#MMMDONE          ;GO TO THE NEXT TEST.

                           ;THIS SUBROUTINE, MODFOV, IS CALLED TO SETUP THE
                           ;OPERANDS, EXECUTE THE MODF INSTRUCTION AND CHECK THE RESULTS.
                           ;IT IS CALLED THUS:
                           ;
                           ;               ACARG:  .WORD   X,X             ;AC OPERAND
                           ;               FSRCARG:.WORD   X,X             ;FSRC OPERAND
                           ;               FRES:   .WORD   X,X             ;FRACTIONAL RESULT
                           ;               INTRES: .WORD   X,X             ;INTEGER RESULT
                           ;               ERFRES: .WORD   X,X             ;ERROR FRACTION RESULT
                           ;               ERINTRES:.WORD  X,X             ;ERROR INTEGER RESULT
                           ;               FPSB:   .WORD   X               ;FPS BEFORE EXECUTION
                           ;               FPSA:   .WORD   X               ;FPS AFTER EXECUTION
                           ;               FEC:    .WORD   X               ;FEC
                           ;               ERR1:   ERROR   X               ;FEC ERROR
                           ;                       BR      CONT
                           ;               ERR2:   ERROR   X               ;INTEGER ERROR
                           ;               CONT:                           ;RETURN ADDRESS
                           ;
                           ;THE OPERANDS ARE SET UP (USING AC0 FOR THE AC ARGUMENT). THE MODF
                           ;INSTRUCTION IS EXECUTED. THEN THE RESULTS ARE RETRIEVED.
                           ;THE FRACTION PART OF THE RESULT IS COMPARED WITH FRES. IF THIS IS CORRECT
                           ;THEN THE INTEGER PART IS COMPARED WITH INTRES. IF BOTH OF THESE ARE CORRECT
                           ;THEN THE FPS IS COMPARED WITH FPSA. AFTER EXECUTION IF NO ERRORS OCCURRED
                           ;THEN MODFOV WILL RETURN TO CONT. IF THE FPS WAS INCORRECT
                           ;IT IS REPORTED HERE. IF THE FRACTION IS INCORRECT IT IS COMPARED WITH
                           ;THE ANTICIPATED BAD FRACTION, ERFRES. IF THIS DOESN'T MATCH
                           ;THE TRUE RESULT THEN THE ERROR IS REPORTED HERE. IF THE ANTICIPATED
                           ;FAILURE MATCHES THE TRUE RESULT THEN MODFOV PASSES CONTROL TO THE
                           ;ERROR CALL AT ERR1. LIKEWISE IF THE INTEGER PART OF THE RESULT IS
                           ;NOT CORRECT THEN IT IS COMPARED WITH THE ANTICIPATED INTEGER
                           ;FAILURE. IF THIS DOEN'T MATCH THEN THE ERROR IS REPORTED HERE.
                           ;IF A MATCH IS MADE HOWEVER, MODFOV WILL RETURN CONTROL TO THE ERROR
                           ;CALL AT ERR2.

077120  012601             MODFOV: MOV     (SP)+,R1           ;GET A POINTER TO THE ARGUMENTS
077122  012700  000200             MOV     #200,R0            ;SET FD MODE.
077126  170100                     LDFPS   R0
077130  010100                     MOV     R1,R0              ;SET UP AC0
077132  172410                     LDD     (R0),AC0
077134  012700  075450             MOV     #MODP1,R0          ;PUT A BACKROUND PATTERN INTO AC1.
077140  172510                     LDD     (R0),AC1
077142  016100  000020             MOV     20(R1),R0          ;SET UP THE FPS.
077146  170100                     LDFPS   R0
077150  010100                     MOV     R1,R0              ;COMPUTE THE ADDRESS OF THE FSRC.
077152  062700  000004             ADD     #4,R0

077156  171410             1$:     MODF    (R0),AC0           ;EXECUTE THE TEST INSTRUCTION.
```

```
077160 170204                              STFPS   R4              ;GET THE FPS.
077162 170305                              STST    R5              ;GET FEC.
077164 012700 000200                       MOV     #200,R0         ;SET FD MODE.
077170 170100                              LDFPS   R0
077172 012700 077314                       MOV     #MODFD0,R0      ;GET THE FRACTIONAL RESULT.
077176 174010                              STD     AC0,(R0)
077200 012700 077324                       MOV     #MODFD1,R0      ;GET THE INTEGER RESULT.
077204 174110                              STD     AC1,(R0)
077206 012702 077314                       MOV     #MODFD0,R2      ;CHECK THE FRACTIONAL RESULT.
077212 026112 000010                       CMP     10(R1),(R2)
077216 001401                              BEQ     2$
077220 104000                              EMT                     ;
077222 026162 000012 000002        2$:     CMP     12(R1),2(R2)
077230 001401                              BEQ     3$
077232 104000                              EMT                     ;
077234 012702 077324               3$:     MOV     #MODFD1,R2      ;CHECK THE INTEGER RESULT.
077240 026112 000014                       CMP     14(R1),(R2)
077244 001401                              BEQ     4$
077246 104000                              EMT                     ;
077250 026162 000016 000002        4$:     CMP     16(R1),2(R2)
077256 001401                              BEQ     5$
077260 104000                              EMT                     ;
077262 026104 000022               5$:     CMP     22(R1),R4       ;CHECK THE FPS.
077266 001401                              BEQ     6$
077270 104000                              EMT                     ;
077272 026105 000024               6$:     CMP     24(R1),R5       ;CHECK THE FEC.
077276 001002                              BNE     25$             ;BRANCH IF INCORRECT.

077300 000161 000030               9$:     JMP     30(R1)          ;RETURN.
                                   ;REPORT FEC ERROR.
077304 010102                      25$:    MOV     R1,R2
077306 062702 000026                       ADD     #26,R2
077312 000112                              JMP     (R2)

077314 000000 000000 000000        MODFD0: .WORD   0,0,0,0
077322 000000

077324 000000 000000 000000        MODFD1: .WORD   0,0,0,0
077332 000000

077334                             MMMDONE:
077334 004767 021134                       JSR     PC,.RSET        ;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?).





                                   ;*******************************************************************
                                   ;TEST 476       UNDER\OVER FLOW, USING MODD WITH TRAPS DISABLED, TEST
                                   ;*******************************************************************
077340                             TS476:

                                   ;UNDERFLOW TEST WITH EXPONENT OF THE RESULT = -129, FIU = 1, FID = 1
```

```
077340  004737  077624           NNN1:   JSR     PC,@#MODDOV
077344  020252  125252           1$:     .WORD   20252,125252     ;AC
077350  125252  125252                   .WORD   125252,125252
077354  020100  000000  000000   2$:     .WORD   20100,0,0,0      ;FSRC
077362  000000
077364  000177  177777  177777   3$:     .WORD   177,-1,-1,-1     ;FRACTIONAL RES.
077372  177777
077374  000000  000000  000000   4$:     .WORD   0,0,0,0          ;INTEGER RES.
077402  000000
077404  042200                   7$:     42200                    ;FPS BEFORE EXECUTION.
077406  142204                           142204                   ;FPS AFTER EXECUTION.
077410  000012                           12                       ;FEC
077412                           8$:
077412  104000                           EMT                      ;
                                 ;UNDERFLOW TEST WITH EXPONENT OF THE RESULT = -193, FIU = 0, FID = 1
077414  004737  077624           NNN2:   JSR     PC,@#MODDOV
077420  010000  000000           1$:     .WORD   10000,0          ;AC
077424  123456  000000                   .WORD   123456,0
077430  010200  000000  000000   2$:     .WORD   10200,0,0,0      ;FSRC
077436  000000
077440  000000  000000  000000   3$:     .WORD   0,0,0,0          ;FRACTIONAL RES.
077446  000000
077450  000000  000000  000000   4$:     .WORD   0,0,0,0          ;INTEGER RES.
077456  000000
077460  005213                   7$.     5213                     ;FPS BEFORE EXECUTION.
077462  005204                           5204                     ;FPS AFTER EXECUTION.
077464  000012                           12
077466  000240                   8$:     NOP
                                 ;OVERFLOW TEST WITH EXPONENT OF THE RESULT = 128, FIV = 1, FID   1
077470  004737  077624           NNN3:   JSR     PC,@#MODDOV
077474  060252  125252           1$:     .WORD   60252,125252     ;AC
077500  125252  125252                   .WORD   125252,125252
077504  060100  000000  000000   2$:     .WORD   60100,0,0,0      ;FSRC
077512  000000
077514  000000  000000  000000   3$:     .WORD   0,0,0,0          ;FRACTIONAL RES.
077522  000000
077524  000177  177777  177777   4$:     .WORD   177,-1,-1,-1     ;INTEGER RES.
077532  177777
077534  041200                   7$:     41200                    ;FPS BEFORE EXECUTION.
077536  141206                           141206                   ;FPS AFTER EXECUTION.
077540  000010                           10                       ;FEC
077542                           8$:
077542  104000                           EMT                      ;
                                 ;OVERFLOW TEST WITH EXPONENT OF THE RESULT = 130, FIV = 0, FID = 1
077544  004737  077624           NNN4:   JSR     PC,@#MODDOV
077550  060200  000000           1$:     .WORD   60200,0          ;AC
077554  125252  000000                   .WORD   125252,0
077560  060200  000000  000000   2$:     .WORD   60200,0,0,0      ;FSRC
077566  000000
077570  000000  000000  000000   3$:     .WORD   0,0,0,0          ;FRACTIONAL RES.
077576  000000
077600  000000  000000  000000   4$:     .WORD   0,0,0,0          ;INTEGER RES.
077606  000000
077610  006211                   7$:     6211                     ;FPS BEFORE EXECUTION.
077612  006206                           6206                     ;FPS AFTER EXECUTION.
077614  000010                           10                       ;FEC
```

```
077616  000240                  8$:     NOP
077620  000137  100040          9$:     JMP     @#NNNDONE       ;GO TO NEXT TEST.

                                ;THIS SUBROUTINE, MODDOV, IS CALLED TO SETUP THE
                                ;OPERANDS, EXECUTE THE MODD INSTRUCTION AND CHECK THE RESULTS.
                                ;IT IS CALLED THUS:
                                ;
                                ;               ACARG:  .WORD   X,X,X,X         ;AC OPERAND
                                ;               FSRCARG:.WORD   X,X,X,X         ;FSRC OPERAND
                                ;               FRES:   .WORD   X,X,X,X         ;FRACTIONAL RESULT
                                ;               INTRES: .WORD   X,X,X,X         ;INTEGER RESULT
                                ;               ERFRES: .WORD   X,X,X,X         ;ERROR FRACTION RESULT
                                ;               ERINTRES:.WORD  X,X,X,X         ;ERROR INTEGER RESULT
                                ;               FPSB:   .WORD   X               ;FPS BEFORE EXECUTION
                                ;               FPSA:   .WORD   X               ;FPS AFTER EXECUTION
                                ;               ERR1:   ERROR   X               ;FRACTION ERROR
                                ;                       BR      CONT
                                ;               ERR2:   ERROR   X               ;INTFGER ERROR
                                ;               CONT:                           ;RETURN ADDRESS
                                ;
                                ;THE OPERANDS ARE SET UP (USING AC0 FOR THE AC ARGUMENT). THE MODD
                                ;INSTRUCTION IS EXECUTED. THEN THE RESULTS ARE RETRIEVED.
                                ;THE FRACTION PART OF THE RESULT IS COMPARED WITH FRES. IF THIS IS CORRECT
                                ;THEN THE INTEGER PART IS COMPARED WITH INTRES. IF BOTH OF THESE ARE CORRECT
                                ;THEN THE FPS IS COMPARED WITH FPSA. AFTER EXECUTION IF NO ERRORS OCCURRED
                                ;THEN MODDOV WILL RETURN TO CONT. IF THE FPS WAS INCORRECT
                                ;IT IS REPORTED HERE. IF THE FRACTION IS INCORRECT IT IS COMPARED WITH
                                ;THE ANTICIPATED BAD FRACTION, ERFRES. IF THIS DOESN'T MATCH
                                ;THE TRUE RESULT THEN THE ERROR IS REPORTED HERE. IF THE ANTICIPATED
                                ;FAILURE MATCHES THE TRUE RESULT THEN  MODDOV PASSES CONTROL TO THE
                                ;ERROR CALL AT ERR1. LIKEWISE IF THE INTEGER PART OF THE RESULT IS
                                ;NOT CORRECT THEN IT IS COMPARED WITH THE ANTICIPATED INTEGER
                                ;FAILURE. IF THIS DOEN'T MATCH THEN THE ERROR IS REPORTED HERE.
                                ;IF A MATCH IS MADE HOWEVER, MODDOV WILL RETURN CONTROL TO THE ERROR
                                ;CALL AT ERR2.

077624  012601                  MODDOV: MOV     (SP)+,R1        ;GET A POINTER TO THE ARGUMENTS
077626  012700  000200                  MOV     #200,R0         ;SET FD MODE.
077632  170100                          LDFPS   R0
077634  010100                          MOV     R1,R0           ;SET UP AC0
077636  172410                          LDD     (R0),AC0
077640  012700  075450                  MOV     #MODP1,R0       ;PUT A BACKROUND PATTERN INTO AC1.
077644  172510                          LDD     (R0),AC1
077646  016100  000040                  MOV     40(R1),R0       ;SET UP THE FPS.
077652  170100                          LDFPS   R0
077654  010100                          MOV     R1,R0           ;COMPUTE THE ADDRESS OF THE FSRC.
077656  062700  000010                  ADD     #10,R0

077662  171410                  1$:     MODD    (R0),AC0        ;EXECUTE THE TEST INSTRUCTION.

077664  170305                          STST    R5              ;GET THE FPS.
077666  170204                          STFPS   R4              ;GET THE FPS.
077670  012700  000200                  MOV     #200,R0         ;SET FD MODE.
077674  170100                          LDFPS   R0
077676  012700  100020                  MOV     #MODDD0,R0      ;GET THE FRACTIONAL RESULT.
077702  174010                          STD     AC0,(R0)
```

```
077704  012700  100030                    MOV     #MODDD1,R0      ;GET THE INTEGER RESULT.
077710  174110                            STD     AC1,(R0)
077712  012702  100020                    MOV     #MODDD0,R2      ;CHECK THE FRACTIONAL RESULT.
077716  010103                            MOV     R1,R3
077720  062703  000020                    ADD     #20,R3
077724  012700  000004                    MOV     #4,R0
077730  022223                    2$:     CMP     (R2)+,(R3)+
077732  001401                            BEQ     4$
077734  104000                            EMT                     ;
077736  077004                    4$:     SOB     R0,2$
077740  012702  100030                    MOV     #MODDD1,R2      ;CHECK THE INTEGER RESULT
077744  010103                            MOV     R1,R3
077746  062703  000030                    ADD     #30,R3
077752  012700  000004                    MOV     #4,R0
077756  022223                    3$:     CMP     (R2)+,(R3)+
077760  001401                            BEQ     5$
077762  104000                            EMT                     ;
077764  077004                    5$:     SOB     R0,3$
077766  026104  000042                    CMP     42(R1),R4       ;CHECK THE FPS.
077772  001401                            BEQ     6$
077774  104000                            EMT                     ;
077776  026105  000044            6$:     CMP     44(R1),R5       ;CHECK THE FEC.
100002  001002                            BNE     25$

100004  000161  000050            9$:     JMP     50(R1)          ;RETURN.
                                  ;REPORT FEC ERROR.
100010  010102                    25$:    MOV     R1,R2
100012  062702  000046                    ADD     #46,R2
100016  000112                            JMP     (R2)

100020  000000  000000  000000    MODDD0: .WORD   0,0,0,0
100026  000000

100030  000000  000000  000000    MODDD1: .WORD   0,0,0,0
100036  000000

100040                            NNNDONE:
100040  004767  020430                    JSR     PC,.RSET        ;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?).



                                  ;***********************************************************************
                                  ;TEST 477       MORE MICROCODES COVERAGE
                                  ;***********************************************************************
100044                            TS477:
100044  012737  100060  000244    XT1:    MOV     #XT1A,@#244
100052  170227  000000                    STFPS   #0
100056  000401                            BR      XT2
100060                            XT1A:
100060  104000                            EMT                     ;

100062  012700  177777            XT2:    MOV     #-1,R0
100066  170127  000000                    LDFPS   #0
```

```
100072 170200                        STFPS   R0
100074 005700                        TST     R0
100076 001401                        BEQ     XT2A
100100 104000                        EMT                     ;
100102 012700 100636          XT2A:  MOV     #XPAT0,R0
100106 172440                        LDF     -(R0),AC0
100110 022700 100632                 CMP     #XPAT0-4,R0
100114 001401                        BEQ     XT2B
100116 104000                        EMT                     ;
100120 170200                 XT2B:  STFPS   R0
100122 022700 000004                 CMP     #4,R0           ;CHECK IF FZ IS SET?
100126 001401                        BEQ     XT3
100130 104000                        EMT                     ;
100132 170127 000000          XT3:   LDFPS   #0
100136 012700 100636                 MOV     #XPAT0,R0
100142 174040                        STF     AC0,-(R0)
100144 022700 100632                 CMP     #XPAT0-4,R0
100150 001401                        BEQ     XT3A
100152 104000                        EMT                     ;
100154 170200                 XT3A:  STFPS   R0
100156 005700                        TST     R0
100160 001401                        BEQ     XT4
100162 104000                        EMT                     ;

100164 170127 000000          XT4:   LDFPS   #0
100170 012737 100214 000244          MOV     #XT4A,@#244
100176 170127 004000                 LDFPS   #04000          ;INTRPT ON UNDEFINED VARIABLE
100202 172437 100636                 LDF     @#XPAT0,AC0
100206 174437 100666                 DIVF    @#XPAT3,AC0     ;GET UNDEFINED VARIABLE. _0
100212 104000                        EMT                     ;
100214 170200                 XT4A:  STFPS   R0
100216 022700 104004                 CMP     #104004,R0      ;CHECK: FER,FIUV,FZ ARE SET?
100222 001401                        BEQ     XT4B
100224 104000                        EMT                     ;
100226 012700 100626          XT4B:  MOV     #XBUF,R0
100232 174010                        STF     AC0,(R0)
100234 005737 100626                 TST     @#XBUF
100240 001401                        BEQ     XT5
100242 104000                        EMT                     ;

100244 012737 100264 000244   XT5:   MOV     #XT5A,@#244
100252 170127 004000                 LDFPS   #04000          ;INTRPT ON UNDEFINED VARIBALE
100256 177437 100666                 LDCDF   @#XPAT3,AC0     ;GET UNDEFINED VARIABLE. _0
100262 104000                        EMT                     ;
100264 170200                 XT5A:  STFPS   R0
100266 022700 104014                 CMP     #104014,R0      ;CHECK: FER,FIUV,FN,FZ ARE SET?
100272 001401                        BEQ     XT5B
100274 104000                        EMT                     ;
100276 012700 100626          XT5B:  MOV     #XBUF,R0
100302 174010                        STF     AC0,(R0)
100304 005737 100626                 TST     @#XBUF
100310 001401                        BEQ     XT6
100312 104000                        EMT                     ;

100314 012737 100340 000244   XT6:   MOV     #XT6A,@#244
100322 170127 004000                 LDFPS   #04000          ;INTRPT ON UNDEFINED VARIBALE
```

```
100326 172437 100636                 LDF     @#XPAT0,AC0
100332 172037 100666                 ADDF    @#XPAT3,AC0
100336 104000                        EMT                     ;
100340 170200                XT6A:   STFPS   R0
100342 022700 104004                 CMP     #104004,R0      ;CHECK: FER,FIUV,FZ ARE SET?
100346 001401                        BEQ     XT6B
100350 104000                        EMT                     ;
100352 012700 100626         XT6B:   MOV     #XBUF,R0
100356 174010                        STF     AC0,(R0)
100360 005737 100626                 TST     @#XBUF
100364 001401                        BEQ     XT7
100366 104000                        EMT                     ;

100370 170127 000000         XT7:    LDFPS   #0
100374 172437 100676                 LDF     @#XPAT4,AC0
100400 175437 100726                 STCFI   AC0,@#XPAT0
100404 022737 000002 100726          CMP     #2,@#XPAT0      ;CHECK DATA
100412 001401                        BEQ     XT8
100414 104000                        EMT                     ;

100416 170127 000100         XT8:    LDFPS   #100            ;SET FL
100422 172437 100676                 LDF     @#XPAT4,AC0
100426 175467 000274                 STCFI   AC0,XPAT0
100432 022737 000002 100730          CMP     #2,@#XPAT0+2
100440 001401                        BEQ     XT9
100442 104000                        EMT                     ;

                             ;START OF FPP2
100444 170127 000000         XT9:    LDFPS   #0
100450 172437 100636                 LDF     @#XPAT0,AC0
100454 172037 100676                 ADDF    @#XPAT4,AC0
100460 170200                        STFPS   R0
100462 005700                        TST     R0
100464 001401                        BEQ     XT10
100466 104000                        EMT                     ;

100470 170127 000000         XT10:   LDFPS   #0
100474 172437 100676                 LDF     @#XPAT4,AC0
100500 173037 100676                 SUBF    @#XPAT4,AC0
100504 170200                        STFPS   R0
100506 022700 000004                 CMP     #4,R0
100512 001401                        BEQ     XT11
100514 104000                        EMT                     ;

100516 170127 000000         XT11:   LDFPS   #0
100522 172437 100676                 LDF     @#XPAT4,AC0
100526 173437 100676                 CMPF    @#XPAT4,AC0
100532 170200                        STFPS   R0
100534 022700 000004                 CMP     #4,R0           ;CHECK IF FZ IS SET?
100540 001401                        BEQ     XT12
100542 104000                        EMT                     ;

100544 170127 000000         XT12:   LDFPS   #0
100550 172437 100676                 LDF     @#XPAT4,AC0
100554 174437 100656                 DIVF    @#XPAT2,AC0
100560 012700 100626                 MOV     #XBUF,R0
```

```
100564  174010                                    STF     AC0,(R0)
100566  022737  040176  100626                    CMP     #040176,@#XBUF     ;CHECK DATA
100574  001401                                    BEQ     XT13
100576  104000                                    EMT                        ;

100600  170127  000000                   XT13:    LDFPS   #0
100604  172437  100706                            LDF     @#XPAT5,AC0
100610  174437  100716                            DIVF    @#XPAT6,AC0
100614  170200                                    STFPS   R0
100616  022700  000004                            CMP     #4,R0
100622  001445                                    BEQ     XTDONE
100624  104000                                    EMT                        ;


100626  000000  000000  000000           XBUF:    .WORD   0,0,0,0
100634  000000
100636  000000  000000  000000           XPAT0:   .WORD   0,0,0,0
100644  000000
100646  000001  000001  000001           XPAT1:   .WORD   1,1,1,1
100654  000001
100656  040401  000000  000000           XPAT2:   .WORD   40401,0,0,0
100664  000000
100666  100000  000000  000000           XPAT3:   .WORD   100000,0,0,0
100674  000000
100676  040400  000000  000000           XPAT4:   .WORD   040400,0,0,0
100704  000000
100706  000207  000000  000000           XPAT5:   .WORD   207,0,0,0
100714  000000
100716  077007  000000  000000           XPAT6:   .WORD   77007,0,0,0
100724  000000
100726  000000  000000  000000           XPAT0:   .WORD   0,0,0,0
100734  000000

100736                                   XTDONE:
100736  004767  017532                            JSR     PC,.RSET           ;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?).




                                         ;*************************************************************************
                                         ;TEST 500        STF WITH ILLEGAL ACCUMULATOR TEST
                                         ;*************************************************************************
100742                                   TS500:

100742  005000                                    CLR     R0                 ;SET THE FPS.
100744  170100                                    LDFPS   R0

100746  012737  100766  000244                    MOV     #OOOT,@#FPVECT     ;SET UP FOR FP TRAPS.
100754  012737  100762  037366                    MOV     #1$,@#$TMP2

100762  174007                           1$:      STF     AC0,AC7            ;THIS TEST INSTRUCTION SHOULD
                                                                             ;CAUSE A TRAP.
```

```
                               ;REPORT FAILURE OF USE OF ILLEGAL ACCUMULATOR 7 TO CAUSE AN FPP TRAP.
100764                         0002:
100764  104000                         EMT                             ;INSTRUCTION DID NOT TRAP

                               ;TRAP TO 000T, HERE, WHEN THE EXPECTED ERROR OCCURS.
100766  011600                 000T:   MOV     (SP),R0                 ;MAKE SURE THE ERROR OCCURRED
100770  022700  100764                 CMP     #0002,R0                ;AT THE CORRECT ADDRESS.
100774  001420                         BEQ     TS501
100776  104000                         EMT                             ;FLOATING POINT TRAP DID NOT OPERATE RIGHT

101000  170204                 0003:   STFPS   R4                      ;GET FPS.
101002  170305                         STST    R5                      ;GET FEC.
101004  012702  100000                 MOV     #100000,R2              ;EXPECTED FPS
101010  012703  000002                 MOV     #2,R3                   ;EXPECTED FEC
101014  022626                         CMP     (SP)+,(SP)+             ;RESET THE STACK.

101016  020204                         CMP     R2,R4                   ;WAS FPS CORRECT?
101020  001401                         BEQ     0004
101022  104000                         EMT                             ;FPS INCORRECTLY SET AFTER USE OF ILLEGAL ACC
101024  020305                 0004:   CMP     R3,R5                   ;WAS THE FEC CORRECT?
101026  001401                         BEQ     000DONE
101030  104000                         EMT                             ;INCORRECT FEC AFTER USE OF ILLEGAL ACC

101032                         000DONE:
101032  004767  017436                 JSR     PC,.RSET                ;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?).




                               ;*************************************************************************
                               ;TEST 501       FDST MODE 1, FLOATING MODE, TEST
                               ;*************************************************************************
101036                         TS501:


101036  012700  177777                 MOV     #-1,R0                  ;SET UP A BACKROUND PATTERN IN THE
101042  012701  101152                 MOV     #PPPBF0,R1              ;INPUT BUFFER.
101046  012702  000014                 MOV     #14,R2
101052  010021                 PPP2:   MOV     R0,(R1)+
101054  077202                         SOB     R2,PPP2

101056  012700  000200                 MOV     #200,R0                 ;SET FD MODE.
101062  170100                         LDFPS   R0
101064  012700  101202                 MOV     #PPPTP1,R0              ;PUT TEST DATA INTO AC0.
101070  172410                         LDD     (R0),AC0

101072  012700  101166                 MOV     #PPPBF1,R0              ;FDST ADDRESS.
101076  005002                         CLR     R2                      ;CLEAR THE FPS.
101100  170102                         LDFPS   R2

101102  174010                 PPP3:   STF     AC0,(R0)                ;TEST INSTRUCTION.
```

```
101104  022700  101166                 CMP     #PPPBF1,R0       ;WAS R0 MODIFIED DURING EXECUTION?
101110  001401                         BEQ     PPP4
101112  104000                         EMT                      ;R0 MODIFIED

101114  012700  101166         PPP4:   MOV     #PPPBF1,R0       ;CHECK THE DATA IN THE OUTPUT BUFFER.
101120  012701  101202                 MOV     #PPPTP1,R1
101124  022021                         CMP     (R0)+,(R1)+
101126  001031                         BNE     PPP10            ;BRANCH IF INCORRECT.
101130  022011                         CMP     (R0)+,(R1)
101132  001027                         BNE     PPP10            ;BRANCH IF INCORRECT.
101134  022720  177777                 CMP     #-1,(R0)+        ;WAS FLOATING MODE USED?
101140  001024                         BNE     PPP10            ;BRANCH IF NOT.
101142  022710  177777                 CMP     #-1,(R0)
101146  001021                         BNE     PPP10
101150  000421                         BR      PPPDONE ;GO TO NEXT TEST.

101152  177777  177777  177777 PPPBF0: .WORD   -1,-1,-1,-1,-1,-1
101160  177777  177777  177777

101166  177777  177777  177777 PPPBF1: .WORD   -1,-1,-1,-1,-1,-1
101174  177777  177777  177777

101202  123456  023456         PPPTP1: .WORD   123456,23456
101206  034567  045671                 .WORD   34567,45671

101212                         PPP10:
101212  104000                         EMT                      ;
101214                         PPPDONE:
101214  004767  017254                 JSR     PC,.RSET         ;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?).





                               ;*************************************************************************
                               ;TEST 502        FDST MODE 2 TEST
                               ;*************************************************************************
101220                         TS502:

                               ;FIRST TEST STF.

101220  012700  177777                 MOV     #-1,R0  ;SET UP THE OUTPUT BUFFER.
101224  012701  101334                 MOV     #QQQBF0,R1
101230  012702  000014                 MOV     #14,R2
101234  010021                 QQQ2:   MOV     R0,(R1)+
101236  077202                         SOB     R2,QQQ2

101240  012700  000200                 MOV     #200,R0          ;SET FD MODE.
101244  170100                         LDFPS   R0
101246  012700  101364                 MOV     #QQQTP1,R0       ;SETUP AC0.
101252  172410                         LDD     (R0),AC0
```

```
101254  012700  101350                   MOV     #QQQBF1,R0      ;FDST ADDRESS.
101260  005002                           CLR     R2
101262  170102                           LDFPS   R2              ;SET FPS.
101264  174020                  QQQ3:    STF     AC0,(R0)+       ;TEST INSTRUCTION.

101266  022700  101354                   CMP     #QQQBF1+4,R0    ;WAS R0 INCREMENTED BY 4 PROPERLY?

101272  001401                           BEQ     QQQ4
101274  104000                           EMT                     ;REPORT R0 INCORRECT AFTER FDST MODE 2
101276  012700  101350          QQQ4:    MOV     #QQQBF1,R0      ;WAS THE OUTPUT DATA CORRECT?
101302  012701  101364                   MOV     #QQQTP1,R1
101306  022021                           CMP     (R0)+,(R1)+
101310  001031                           BNE     QQQ10           ;BRANCH IF INCORRECT.
101312  022021                           CMP     (R0)+,(R1)+
101314  001027                           BNE     QQQ10           ;BRANCH IF INCORRECT.
101316  022027  177777                   CMP     (R0)+,#-1       ;SEE IF ANY OTHER DATA BUFFER WORDS WERE MODIFIED.
101322  001024                           BNE     QQQ10           ;BRANCH IF INCORRECT.
101324  022027  177777                   CMP     (R0)+,#-1
101330  001021                           BNE     QQQ10           ;BRANCH IF INCORRECT.
101332  000421                           BR      QQQ20
101334  177777  177777  177777  QQQBF0:  .WORD   -1,-1,-1,-1,-1,-1
101342  177777  177777  177777
101350  177777  177777  177777  QQQBF1:  .WORD   -1,-1,-1,-1,-1,-1
101356  177777  177777  177777
101364  076543                  QQQTP1:  76543
101366  065432                           65432
101370  054321                           54321
101372  043210                           43210
                                ;REPORT OUTPUT DATA INCORRECT:
101374                          QQQ10:
101374  104000                           EMT                     ;

                                ;NOW TEST STD MODE 2.

101376  012700  101334          QQQ20:   MOV     #QQQBF0,R0      ;SET UP DEFAULT INPUT DATA BUFFER.
101402  010001                           MOV     R0,R1
101404  012702  000014                   MOV     #14,R2
101410  010021                  QQQ22:   MOV     R0,(R1)+
101412  077202                           SOB     R2,QQQ22
101414  012700  000200                   MOV     #200,R0         ;ENTER FLOATING DOUBLE MODE.
101420  170100                           LDFPS   R0
101422  012700  101364                   MOV     #QQQTP1,R0      ;LOAD AC0.
101426  172410                           LDD     (R0),AC0
101430  012700  101350                   MOV     #QQQBF1,R0      ;SET DESTINATION ADDRESS.
101434  012737  101442  037366           MOV     #QQQ23,@#$TMP2
101442  174020                  QQQ23:   STD     AC0,(R0)+       ;TEST INSTRUCTION.
101444  022700  101360                   CMP     #QQQBF1+10,R0   ;WAS R0 INCREMENTED BY 10 CORRECTLY?
101450  001401                           BEQ     QQQ24
101452  104000                           EMT                     ;REPORT R0 INCORRECTLY INCREMENTED
101454  012700  101350          QQQ24:   MOV     #QQQBF1,R0      ;DID THE DATA REACH THE OUTPUT BUFFER CORRECTLY?
101460  012701  101364                   MOV     #QQQTP1,R1
101464  012702  000004                   MOV     #4,R2
101470  022021                  1$:      CMP     (R0)+,(R1)+
101472  001002                           BNE     QQQ25           ;BRANCH IF INCORRECT.
101474  077203                           SOB     R2,1$
101476  000401                           BR      QQQDONE
```

```
                                  ;REPORT DATA INCORRECT.
101500                            QQQ25:
101500  104000                            EMT                         ;
101502                            QQQDONE:
101502  004767  016766                    JSR     PC,.RSET            ;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?).

                                  ;*******************************************************************************
                                  ;TEST 503       FDST MODE 2, WITH GR7, TEST
                                  ;*******************************************************************************
101506                            TS503:

101506  012700  101556                    MOV     #RRR3,R0            ;SET UP THE DATA BUFFER FOLLOWING THE TEST INSTRUCTION.
101512  012701  101624                    MOV     #RRRTP1,R1
101516  012702  000004                    MOV     #4,R2
101522  012021                    1$:     MOV     (R0)+,(R1)+
101524  077202                            SOB     R2,1$
101526  012700  000200                    MOV     #200,R0             ;ENTER FLOATING DOUBLE MODE.
101532  170100                            LDFPS   R0
101534  012700  101634                    MOV     #RRRTP2,R0          ;SET UP AC0.
101540  172410                            LDD     (R0),AC0
101542  012737  101622  000004            MOV     #RRR10,@#ERRVECT ;SET UP FOR AN ODD ADDRESS.
101550  005001                            CLR     R1
101552  005004                            CLR     R4
                                  ;THIS IS THE TEST INSTRUCTION. IT SHOULD MODIFY THE FIRST LOCATION
                                  ;AFTER IT TO BE AN INCREMENT R4, INC R4, INSTRUCTION INSTEAD
                                  ;OF AN INCREMENT R1 INSTRUCTION. THE INCREMENT R4 SHOULD NOT BE
                                  ;EXECUTED SINCE THE PC SHOULD BE INCREMENTED BY TWO DURING IMMEDIATE
                                  ;MODE ADDRESSING. THUS AFTER THE EXECUTION OF THE NEXT 5 INSTRUCTIONS
                                  ;R1 SHOULD CONTAIN 3 AND R4 SHOULD CONTAIN 0.
101554  174027                    RRR2:   STD     AC0,(R7)+           ;TEST INSTRUCTION.
101556  005201                    RRR3:   INC     R1                  ;THE STD INSTRUCTION SHOULD CHANGE THIS TO INC R4.
101560  005201                            INC     R1
101562  005201                            INC     R1
101564  005201                            INC     R1
101566  012700  101644                    MOV     #RRREXP,R0          ;SEE IF THE DATA WAS OUTPUT CORRECTLY.
101572  012702  101556                    MOV     #RRR3,R2
101576  012703  000004                    MOV     #4,R3
101602  022022                    RRR4:   CMP     (R0)+,(R2)+
101604  001006                            BNE     RRR10               ;BRANCH IF INCORRECT.
101606  077303                            SOB     R3,RRR4
101610  005704                            TST     R4                  ;MAKE SURE R4 IS 0.
101612  001003                            BNE     RRR10               ;BRANCH IF R4 IS INCORRECT.
101614  022701  000003                    CMP     #3,R1               ;SEE IF R1 IS CORRECT.
101620  001415                            BEQ     RRRDONE
101622                            RRR10:
101622  104000                            EMT                         ;
                                  ;THESE ARE TEST DATA PATTERNS USED TO SET UP THE OUTPUT BUFFER AT RRR3.
101624  005201                    RRRTP1: INC     R1
101626  005201                            INC     R1
101630  005201                            INC     R1
101632  005201                            INC     R1
                                  ;THIS IS THE DATA PUT IN AC0 BEFORE EXECUTION OF THE STD.
```

```
101634  005204                  RRRTP2: INC     R4
101636  005204                          INC     R4
101640  005204                          INC     R4
101642  005204                          INC     R4
                                ;THIS IS THE EXPECTED DATA AT RRR3 AFTER EXECUTION OF THE STD.
101644  005204                  RRREXP: INC     R4
101646  005201                          INC     R1
101650  005201                          INC     R1
101652  005201                          INC     R1
101654                          RRRDONE:
101654  004767  016614                  JSR     PC,.RSET        ;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?).

                                ;*******************************************************************************
                                ;TEST 504       FDST MODE 4 TEST
                                ;*******************************************************************************
101660                          TS504:

101660  012700  177777                  MOV     #-1,R0          ;SET UP THE OUTPUT BUFFER.
101664  012701  102006                  MOV     #SSSBFO,R1
101670  012702  000010                  MOV     #10,R2
101674  010021                  1$:     MOV     R0,(R1)+
101676  077202                          SOB     R2,1$
101700  012700  000200                  MOV     #200,R0         ;ENTER FLOATING DOUBLE MODE.
101704  170100                          LDFPS   R0
101706  012700  102026                  MOV     #SSSTP1,R0      ;SET UP AC0.
101712  172410                          LDD     (R0),AC0
101714  012737  102036  000004          MOV     #SSS10,@#ERRVECT ;SET UP FOR A TRAP TO 4.
101722  012700  102016                  MOV     #SSSA1,R0       ;SET UP THE DESTINATION ADDRESS.

101726  174040                  SSS2:   STD     AC0,-(R0)       ;TEST INSTRUCTION.
101730  005201                          INC     R1
101732  020027  102006                  CMP     R0,#SSSBFO      ;SEE IF R0 WAS DECREMENTED PROPERLY.
101736  001037                          BNE     SSS10           ;BRANCH IF R0 IS INCORRECT.
101740  012700  102006                  MOV     #SSSBFO,R0      ;WAS THE OUTPUT DATA CORRECT?
101744  012701  102026                  MOV     #SSSTP1,R1
101750  012702  000004                  MOV     #4,R2
101754  022021                  1$:     CMP     (R0)+,(R1)+
101756  001027                          BNE     SSS10           ;BRANCH IF INCORRECT.
101760  077203                          SOB     R2,1$
101762  012700  177777                  MOV     #-1,R0          ;IS THE REST OF THE OUTPUT BUFFER CORRECT, -1?
101766  012701  102016                  MOV     #SSSA1,R1
101772  012702  000004                  MOV     #4,R2
101776  020021                  2$:     CMP     R0,(R1)+
102000  001016                          BNE     SSS10           ;BRANCH IF INCORRECT.
102002  077203                          SOB     R2,2$
102004  000415                          BR      SSSDONE

                                ;THIS IS THE OUTPUT DATA BUFFER.
102006  177777                  SSSBFO: -1
102010  177777                          -1
102012  177777                          -1
102014  177777                          -1
```

```
102016  177777                     SSSA1:  -1
102020  177777                             -1
102022  177777                             -1
102024  177777                             -1

                                   ;THIS IS THE TEST DATA LOADED INTO AC0:
102026  147250                     SSSTP1: 147250
102030  036147                             36147
102032  025036                             25036
102034  147250                             147250

102036                             SSS10:
102036  104000                             EMT                             ;
102040                             SSSDONE:
102040  004767  016430                     JSR     PC,.RSET        ;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?).

                                   ;*************************************************************************
                                   ;TEST 505        FDST MODE 3 TEST
                                   ;*************************************************************************
102044                             TS505:

102044  012701  102154                     MOV     #TTTBF0,R1      ;SET UP THE OUTPUT DATA BUFFER.
102050  012700  177777                     MOV     #-1,R0
102054  012702  000013                     MOV     #13,R2
102060  010021                     1$:     MOV     R0,(R1)+
102062  077202                             SOB     R2,1$
102064  012737  102154  102170             MOV     #TTTBF0,@#TTTA2
102072  012700  000200                     MOV     #200,R0         ;ENTER DOUBLE FLOATING MODE.
102076  170100                             LDFPS   R0
102100  012700  102172                     MOV     #TTTTP1,R0      ;SET UP AC0.
102104  172410                             LDD     (R0),AC0
102106  012737  102202  000004             MOV     #TTT10,@#ERRVECT ;SET UP FOR TRAPS TO 4.
102114  012700  102170                     MOV     #TTTA2,R0       ;SET UP THE DESTINATION ADDRESS.

102120  174030                     TTT2:   STD     AC0,@(R0)+      ;TEST INSTRUCTION.

102122  020027  102172                     CMP     R0,#TTTA2+2     ;SEE IF R0 WAS INCREMENTED CORRECTLY.
102126  001025                             BNE     TTT10           ;BRANCH IF INCORRECT.
102130  012701  102154                     MOV     #TTTBF0,R1      ;CHECK THE OUTPUT DATA BUFFER.
102134  012702  102172                     MOV     #TTTTP1,R2
102140  012703  000004                     MOV     #4,R3
102144  022122                     TTT3:   CMP     (R1)+,(R2)+
102146  001015                             BNE     TTT10           ;BRANCH IF NOT CORRECT.
102150  077303                             SOB     R3,TTT3
102152  000414                             BR      TTTDONE

                                   ;THIS IS THHE OUTPUT DATA BUFFER:
102154  177777                     TTTBF0: -1
102156  177777                             -1
102160  177777                             -1
102162  177777                             -1
102164  177777                             -1
```

```
102166  177777                      TTTA1:  -1
102170  102154                      TTTA2:  TTTBF0
102172  101213                      TTTTP1: 101213
102174  141516                              141516
102176  071727                              71727
102200  037475                              37475

102202                              TTT10:
102202  104000                              EMT                             ;

102204                              TTTDONE:
102204  004767  016264                      JSR     PC,.RSET        ;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?).

                                    ;*******************************************************************************
                                    ;TEST 506       FDST MODE 5 TEST
                                    ;*******************************************************************************
102210                              TS506:

102210  012701  102320                      MOV     #UUUBF0,R1      ;SET UP THE OUTPUT DATA BUFFER.
102214  012700  177777                      MOV     #-1,R0
102220  012702  000013                      MOV     #13,R2
102224  010021                      1$:     MOV     R0,(R1)+
102226  077202                              SOB     R2,1$
102230  012737  102320  102332              MOV     #UUUBF0,@#UUUA1
102236  012700  000200                      MOV     #200,R0         ;ENTER DOUBLE FLOATING MODE.
102242  170100                              LDFPS   R0
102244  012700  102336                      MOV     #UUUTP1,R0      ;SET UP AC0.
102250  172410                              LDD     (R0),AC0
102252  012737  102346  000004              MOV     #UUU10,@#ERRVECT ;GET READY FOR ANY TRAPS TO 4.
102260  012700  102334                      MOV     #UUUA2,R0       ;SET UP THE DESTINATION ADDRESS.
102264  174050                      UUU2:   STD     AC0,@-(R0)      ;TEST INSTRUCTION.
102266  020027  102332                      CMP     R0,#UUUA2-2     ;WAS R0 DECRIMENTED PROPERLY?
102272  001025                              BNE     UUU10           ;BRANCH IF R0 IS INCORRECT.
102274  012701  102320                      MOV     #UUUBF0,R1      ;WAS THE DATA OUTPUT CORRECTLY?
102300  012702  102336                      MOV     #UUUTP1,R2
102304  012703  000004                      MOV     #4,R3
102310  022122                      UUU3:   CMP     (R1)+,(R2)+
102312  001015                              BNE     UUU10           ;BRANCH IF DATA IS INCORRECT.
102314  077303                              SOB     R3,UUU3
102316  000414                              BR      UUUDONE

                                    ;THIS IS THE OUTPUT DATA BUFFER
102320  177777                      UUUBF0: -1
102322  177777                              -1
102324  177777                              -1
102326  177777                              -1
102330  177777                              -1
102332  102320                      UUUA1:  UUUBF0
102334  177777                      UUUA2:  -1
102336  020212                      UUUTP1: 20212
102340  023242                              23242
102342  026273                              26273
```

```
102344  031323                                  031323

102346                          UUU10:
102346  104000                          EMT                             ;
102350                          UUUDONE:
102350  004767  016120                  JSR     PC,.RSET                ;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?).

                                ;*******************************************************************************
                                ;TEST 507       FDST MODE 6, INDEX MODE, TEST
                                ;*******************************************************************************
102354                          TS507:

102354  012700  000200                  MOV     #200,R0                 ;ENTER DOUBLE FLOATING MODE.
102360  170100                          LDFPS   R0
102362  012701  102464                  MOV     #VVVBFO,R1              ;SET UP THE OUT PUT DATA BUFFER.
102366  012700  177777                  MOV     #-1,R0
102372  012702  000004                  MOV     #4,R2
102376  010021                  1$:     MOV     R0,(R1)+
102400  077202                          SOB     R2,1$
102402  012737  102504  000004          MOV     #VVV10,@#ERRVECT        ;SET UP VECTOR 4 INCASE OF ERROR.
102410  012700  102474                  MOV     #VVVTP1,R0              ;SET UP AC0.
102414  172410                          LDD     (R0),AC0
102416  012700  074563                  MOV     #VVVBFO-5701,R0         ;SET UP THE DESTINATION ADDRESS.
102422  012701  000001                  MOV     #1,R1
102426  174060  005701          VVV2:   STD     AC0,5701(R0)            ;TEST INSTRUCTION.

102432  020027  074563                  CMP     R0,#VVVBFO-5701         ;SEE IF R0 WAS MODIFIED.
102436  001022                          BNE     VVV10                   ;BRANCH IF INCORRECT.
102440  012702  102464                  MOV     #VVVBFO,R2              ;WAS THE OUTPUT DATA CORRECT.
102444  012703  102474                  MOV     #VVVTP1,R3
102450  012704  000004                  MOV     #4,R4
102454  022223                  1$:     CMP     (R2)+,(R3)+
102456  001012                          BNE     VVV10                   ;BRANCH IF INCORRECT DATA.
102460  077403                          SOB     R4,1$
102462  000411                          BR      VVVDONE
102464  177777                  VVVBFO: -1
102466  177777                          -1
102470  177777                          -1
102472  177777                          -1
102474  030313                  VVVTP1: 30313
102476  023334                          23334
102500  035363                          35363
102502  074041                          74041

102504                          VVV10:
102504  104000                          EMT                             ;
102506                          VVVDONE:
102506  004767  015762                  JSR     PC,.RSET                ;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?).
```

```
                                        ;*************************************************************************
                                        ;TEST 510        FDST MODE 7, INDEX DEFERRED MODE, TEST
                                        ;*************************************************************************
102512                                  TS510:

102512  012700  000200                          MOV     #200,R0         ;ENTER DOUBLE FLOATING MODE.
102516  170100                                  LDFPS   R0
102520  012701  102630                          MOV     #WWWBF0,R1      ;SET UP THE OUTPUT DATA BUFFER.
102524  012700  177777                          MOV     #-1,R0
102530  012702  000004                          MOV     #4,R2
102534  010021                          1$:     MOV     R0,(R1)+
102536  077202                                  SOB     R2,1$
102540  012737  102660  000004                  MOV     #WWW10,@#ERRVECT ;SET UP FOR TRAPS TO 4.
102546  012700  102640                          MOV     #WWWTP1,R0      ;SET UP AC0.
102552  172410                                  LDD     (R0),AC0
102554  012700  074747                          MOV     #WWWBF1-5701,R0 ;SET UP THE DESTINATION ADDRESS.
102560  012701  000001                          MOV     #1,R1
102564  012737  102630  102650                  MOV     #WWWBF0,@#WWWBF1
102572  174070  005701                  WWW2:   STD     AC0,@5701(R0)   ;TEST INSTRUCTION.

102576  020027  074747                          CMP     R0,#WWWBF1-5701 ;IS R0 CORRECT?
102602  001026                                  BNE     WWW10           ;BRANCH IF INCORRECT.
102604  012702  102630                          MOV     #WWWBF0,R2      ;WAS THE DATA OUTPUT CORRECTLY?
102610  012703  102640                          MOV     #WWWTP1,R3
102614  012704  000004                          MOV     #4,R4
102620  022223                          1$:     CMP     (R2)+,(R3)+
102622  001016                                  BNE     WWW10           ;BRANCH IF DATA IS INCORRECT.
102624  077403                                  SOB     R4,1$
102626  000415                                  BR      WWWDONE
102630  177777                          WWWBF0: -1
102632  177777                                  -1
102634  177777                                  -1
102636  177777                                  -1
102640  041424                          WWWTP1: 41424
102642  034445                                  34445
102644  046475                                  46475
102646  051525                                  051525
102650  177777                          WWWBF1: -1
102652  177777                                  -1
102654  177777                                  -1
102656  177777                                  -1

102660                                  WWW10:
102660  104000                                  EMT                     ;
102662                                  WWWDONE:
102662  004767  015606                          JSR     PC,.RSET        ;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?).

                                        ;*************************************************************************
                                        ;TEST 511        STCFD TEST
                                        ;*************************************************************************
102666                                  TS511:
```

```
                                    ;AC=0
102666  004767  000262              XXX1:   JSR     PC,STCFDS
102672  000000                      1$:     0               ;AC
102674  000000                              0
102676  000000                              0
102700  000000                              0
102702  000000                      2$:     0               ;RES
102704  000000                              0
102706  000000                              0
102710  000000                              0
102712  000000                      3$:     0               ;ERROR RES.
102714  000000                              0
102716  177777                              -1
102720  177777                              -1
102722  047000                      4$:     47000           ;FPS BEFORE EXECUTION.
102724  047004                              47004           ;FPS AFTER EXECUTION.
102726  177777                              -1              ;FEC
102730  147004                              147004          ;ERROR FPS.


102732  004767  000216              XXX2:   JSR     PC,STCFDS
102736  017203                      1$:     17203           ;AC
102740  142536                              142536
102742  047506                              47506
102744  172031                              172031
102746  017203                      2$:     17203           ;RES
102750  142536                              142536
102752  000000                              0
102754  000000                              0
102756  017203                      3$:     17203           ;ERROR RES.
102760  142536                              142536
102762  047506                              47506
102764  172031                              172031
102766  040000                      4$:     40000           ;FPS BEFORE EXECUTION.
102770  040000                              40000           ;FPS AFTER EXECUTION.
102772  177777                              -1              ;FEC
102774  177777                              -1              ;ERROR FPS.


102776  004767  000152              XXX3:   JSR     PC,STCFDS
103002  050717                      1$:     50717           ;AC
103004  027374                              27374
103006  075767                              75767
103010  077071                              77071
103012  050717                      2$:     50717           ;RES
103014  027374                              27374
103016  000000                              0
103020  000000                              0
103022  000000                      3$:     0               ;ERROR RES.
103024  000000                              0
103026  000000                              0
103030  000000                              0
103032  047000                      4$:     47000           ;FPS BEFORE EXECUTION.
103034  047000                              47000           ;FPS AFTER EXECUTION.
103036  177777                              -1              ;FEC
```

```
103040  174002                    174002          ;ERROR FPS.


103042  004767  000106    XXX4:   JSR     PC,STCFDS
103046  020212            1$:     20212           ;AC
103050  032425                    32425
103052  026272                    26272
103054  002123                    02123
103056  020212            2$:     20212           ;RES
103060  032425                    32425
103062  000000                    0
103064  000000                    0
103066  020212            3$:     20212           ;ERROR RES.
103070  032425                    32425
103072  100000                    100000
103074  000000                    0
103076  040000            4$:     40000           ;FPS BEFORE EXECUTION.
103100  040000                    40000           ;FPS AFTER EXECUTION.
103102  177777                    -1              ;FEC
103104  177777                    -1              ;ERROR FPS.


103106  004767  000042    XXX5:   JSR     PC,STCFDS
103112  121314            1$:     121314          ;AC
103114  151617                    151617
103116  101112                    101112
103120  131415                    131415
103122  121314            2$:     121314          ;RES
103124  151617                    151617
103126  000000                    0
103130  000000                    0
103132  021314            3$:     21314           ;ERROR RES.
103134  151617                    151617
103136  000000                    0
103140  000000                    0
103142  040000            4$:     40000           ;FPS BEFORE EXECUTION.
103144  040010                    40010           ;FPS AFTER EXECUTION.
103146  177777                    -1              ;FEC
103150  177777                    -1              ;ERROR FPS.
103152  000460            6$:     BR      XXXDONE

                          ;


                          ;THIS SUBROUTINE, STCFDS, IS USED TO SET UP THE OPERANDS, EXECUTE
                          ;THE STCFD INSTRUCTION AND CHECK THE RESULTS. A CALL
                          ;TO IT IS MADE THUS:
                          ;
                          ;               JSR     PC,@#STCFDS
                          ;               ACARG:  .WORD   X,X,X,X         ;AC OPERAND
                          ;               RES:    .WORD   X,X,X,X         ;EXPECTED RESULT
                          ;               ERRES:  .WORD   X,X,X,X         ;ERROR RESULT
                          ;               FPSB:   .WORD   X               ;FPS BEFORE EXECUTION
                          ;               FPSA:   .WORD   X               ;FPS AFTER EXECUTION
                          ;               FEC:    .WORD   X               ;EXPECTED FEC
                          ;               ERFPS:  .WORD   X               ;ERROR FPS.
```

```
                                ;               ERR1:   ERROR   X                       ;DATA ERROR.
                                ;                       BR      CONT
                                ;               ERR2:   ERROR   X                       ;FPS ERROR.
                                ;               CONT:                                   ;RETURN ADDRESS
                                ;
                                ;THE OPERANDS ARE SET UP (USING AC0 AS THE ACCUMULATOR). THEN
                                ;THE STCFD INSTRUCTION IS EXECUTED.
                                ;THE RESULT IS CHECKED AGAINST RES. IF THE RESULT IS CORRECT THEN THE FPS IS
                                ;COMPARED WITH FPSA IF THIS TOO IS CORRECT STCFDS RETURNS CONTROL
                                ;TO THE CALLING ROUTINE AT CONT. IF THE FPS IS BAD STCFDS
                                ;COMPARE IT TO ERROR FPS. IF THIS MATCHES THEN STCFDS WILL RETURN
                                ;TO THE ERROR CALL AT ERR2, OTHERWISE STCFDS ITSELF
                                ;REPORTS THIS FAILURE AND THEN RETURNS TO CONT. IF THE RESULT OF THE
                                ;STCFD IS INCORRECT, THE INCORRECT RESULT IS COMPARED WITH THE
                                ;ANTICIPATED FAILING DATA PATTERN, ERRES. IF THE FAILURE IN
                                ;THE RESULT WAS ANTICIPATED CORRECTLY TO BE ERRES THEN STCFDS
                                ;WILL TRANSFER CONTROL TO THE ERROR CALL AT ERR1. OTHERWISE THE
                                ;RESULT WAS INCORRECT BUT WAS NOT ANTICIPATED AND STCFDS WILL
                                ;REPORT THE FAILURE AFTER WHICH CONTROL WILL BE PASSED TO CONT.

103154  012601                  STCFDS: MOV     (SP)+,R1                ;PICK UP THE POINTER TO THE OPERANDS.
103156  012700  000200                  MOV     #200,R0         ;ENTER DOUBLE FLOATING MODE.
103162  170100                          LDFPS   R0
103164  010100                          MOV     R1,R0           ;LOAD AC0.
103166  172410                          LDD     (R0),AC0
103170  012700  177777                  MOV     #-1,R0          ;FILL THE OUTPUT BUFFER WITH -1'S.
103174  012702  103304                  MOV     #STCFT,R2
103200  012703  000004                  MOV     #4,R3
103204  010022                  1$:     MOV     R0,(R2)+
103206  077302                          SOB     R3,1$
103210  016100  000030                  MOV     30(R1),R0       ;LOAD THE FPS.
103214  170100                          LDFPS   R0
103216  012700  103304                  MOV     #STCFT,R0       ;SET UP THE DESTINATION ADDRESS.
103222  176010                  2$:     STCFD   AC0,(R0)        ;TEST INSTRUCTION.

103224  170204                          STFPS   R4              ;GET THE FPS.
103226  170305                          STST    R5              ;GET THE FEC.
103230  010102                          MOV     R1,R2           ;CHECK THE RESULT.
103232  062702  000010                  ADD     #10,R2
103236  012703  103304                  MOV     #STCFT,R3
103242  012700  000004                  MOV     #4,R0
103246  022223                  3$:     CMP     (R2)+,(R3)+
103250  001014                          BNE     10$             ;BRANCH IF INCORRECT.
103252  077003                          SOB     R0,3$

103254  016102  000032                  MOV     32(R1),R2
103260  020204                          CMP     R2,R4           ;IS THE FPS CORRECT?
103262  001007                          BNE     10$             ;BRANCH IF FPS INCORRECT.
103264  005702                          TST     R2              ;IF EXPECTED FPS IS NEGATIVE, THEN
103266  100003                          BPL     4$              ;GO AHEAD AND CHECK THE FEC.
103270  026105  000036                  CMP     36(R1),R5
103274  001002                          BNE     10$             ;BRANCH IF FEC IS INCORRECT.
103276  000161  000040          4$:     JMP     40(R1)          ;RETURN.
103302                          10$:
103302  104000                          EMT                     ;
103304  177777  177777  177777  STCFT:  -1,-1,-1,-1
```

```
103312  177777
103314                          XXXDONE:
103314  004767  015154                  JSR     PC,.RSET        ;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?).

                                ;*************************************************************************
                                ;TEST 512        STCDF TEST
                                ;*************************************************************************
103320                          TS512:

                                ;AC=0
103320  004767  000262          YYY1:   JSR     PC,STCDFS
103324  000000                  1$:     0                       ;AC
103326  000000                          0
103330  000000                          0
103332  000000                          0
103334  000000                  2$:     0                       ;RES
103336  000000                          0
103340  177777                          -1
103342  177777                          -1
103344  000000                  3$:     0                       ;ERROR RES.
103346  000000                          0
103350  000000                          0
103352  000000                          0
103354  047200                  4$:     47200                   ;FPS BEFORE EXECUTION.
103356  047204                          47204                   ;FPS AFTER EXECUTION.
103360  177777                          -1                      ;FEC
103362  177777                          -1                      ;ERROR FPS.


103364  004767  000216          YYY2:   JSR     PC,STCDFS
103370  067574                  1$:     67574                   ;AC0
103372  073727                          73727
103374  170777                          170777
103376  067574                          67574
103400  067574                  2$:     67574                   ;RES
103402  073730                          73730
103404  177777                          -1
103406  177777                          -1
103410  067574                  3$:     67574                   ;ERROR RES.
103412  073727                          73727
103414  177777                          -1
103416  177777                          -1
103420  040200                  4$:     40200                   ;FPS BEFORE EXECUTION.
103422  040200                          40200                   ;FPS AFTER EXECUTION.
103424  177777                          -1                      ;FEC
103426  177777                          -1                      ;ERROR FPS.


103430  004767  000152          YYY3:   JSR     PC,STCDFS
103434  077777                  1$:     77777                   ;AC0
103436  177777                          -1
103440  100000                          100000
```

```
103442  000000                          0
103444  000000                  2$:     0                       ;RES
103446  000000                          0
103450  177777                          -1
103452  177777                          -1
103454  077777                  3$:     77777                   ;ERROR RES.
103456  177777                          -1
103460  177777                          -1
103462  177777                          -1
103464  040200                  4$:     40200                   ;FPS BEFORE EXECUTION.
103466  040206                          40206                   ;FPS AFTER EXECUTION.
103470  177777                          -1                      ;FEC
103472  040204                          40204                   ;ERROR FPS.


103474  004767  000106          YYY4:   JSR     PC,STCDFS
103500  077777                  1$:     77777                   ;AC0
103502  177777                          -1
103504  100000                          100000
103506  000000                          0
103510  000000                  2$:     0                       ;RES
103512  000000                          0
103514  177777                          -1
103516  177777                          -1
103520  077777                  3$:     77777                   ;ERROR RES.
103522  177777                          -1
103524  177777                          -1
103526  177777                          -1
103530  040200                  4$:     40200                   ;FPS BEFORE EXECUTION.
103532  040206                          40206                   ;FPS AFTER EXECUTION.
103534  177777                          -1                      ;FEC
103536  140206                          140206                  ;ERROR FPS.


103540  004767  000042          YYY5:   JSR     PC,STCDFS
103544  177777                  1$:     177777                  ;AC0
103546  177777                          -1
103550  100000                          100000
103552  000000                          0
103554  100000                  2$:     100000                  ;RES
103556  000000                          0
103560  177777                          -1
103562  177777                          -1
103564  000000                  3$:     0                       ;ERROR RES.
103566  000000                          0
103570  177777                          -1
103572  177777                          -1
103574  047200                  4$:     47200                   ;FPS BEFORE EXECUTION.
103576  147216                          147216                  ;FPS AFTER EXECUTION.
103600  000010                          10                      ;FEC
103602  047206                          47206                   ;ERROR FPS.
103604  000460                  6$:     BR      YYYDONE
                                ;THIS SUBROUTINE, STCDFS, IS USED TO SET UP THE OPERANDS, EXECUTE
                                ;THE STCDF INSTRUCTION AND CHECK THE RESULTS. A CALL
                                ;TO IT IS MADE THUS:
                                ;
```

```
                                 ;               JSR     PC,@#STCFDS
                                 ;               ACARG:  .WORD   X,X,X,X         ;AC OPERAND
                                 ;               RES:    .WORD   X,X,X,X         ;EXPECTED RESULT
                                 ;               ERRES:  .WORD   X,X,X,X         ;ERROR RESULT
                                 ;               FPSB:   .WORD   X               ;FPS BEFORE EXECUTION
                                 ;               FPSA:   .WORD   X               ;FPS AFTER EXECUTION
                                 ;               FEC:    .WORD   X               ;EXPECTED FEC
                                 ;               ERFPS:  .WORD   X               ;ERROR FPS.
                                 ;               ERR1:   ERROR   X               ;DATA ERROR.
                                 ;                       BR      CONT
                                 ;               ERR2:   ERROR   X               ;FPS ERROR.
                                 ;               CONT:                           ;RETURN ADDRESS
                                 ;
                                 ;THE OPERANDS ARE SET UP (USING AC0 AS THE ACCUMULATOR). THEN
                                 ;THE STCFD INSTRUCTION IS EXECUTED.
                                 ;THE RESULT IS CHECKED AGAINST RES. IF THE RESULT IS CORRECT THEN THE FPS IS
                                 ;COMPARED WITH FPSA IF THIS TOO IS CORRECT STCFDS RETURNS CONTROL
                                 ;TO THE CALLING ROUTINE AT CONT. IF THE FPS IS BAD STCFDS
                                 ;COMPARE IT TO ERROR FPS. IF THIS MATCHES THEN STCFDS WILL RETURN
                                 ;TO THE ERROR CALL AT ERR2, OTHERWISE STCFDS ITSELF
                                 ;REPORTS THIS FAILURE AND THEN RETURNS TO CONT. IF THE RESULT OF THE
                                 ;STCFD IS INCORRECT, THE INCORRECT RESULT IS COMPARED WITH THE
                                 ;ANTICIPATED FAILING DATA PATTERN, ERRES. IF THE FAILURE IN
                                 ;THE RESULT WAS ANTICIPATED CORRECTLY TO BE ERRES THEN STCFDS
                                 ;WILL TRANSFER CONTROL TO THE ERROR CALL AT ERR1. OTHERWISE THE
                                 ;RESULT WAS INCORRECT BUT WAS NOT ANTICIPATED AND STCFDS WILL
                                 ;REPORT THE FAILURE AFTER WHICH CONTROL WILL BE PASSED TO CONT.

103606  012601                   STCDFS: MOV     (SP)+,R1        ;PICK UP THE POINTER TO THE OPERANDS.
103610  012700  000200                   MOV     #200,R0         ;ENTER DOUBLE FLOATING MODE.
103614  170100                           LDFPS   R0
103616  010100                           MOV     R1,R0           ;LOAD AC0.
103620  172410                           LDD     (R0),AC0
103622  012700  177777                   MOV     #-1,R0          ;FILL THE OUTPUT BUFFER WITH -1'S.
103626  012702  103736                   MOV     #STCDT,R2
103632  012703  000004                   MOV     #4,R3
103636  010022                   1$:     MOV     R0,(R2)+
103640  077302                           SOB     R3,1$
103642  016100  000030                   MOV     30(R1),R0       ;LOAD THE FPS.
103646  170100                           LDFPS   R0
103650  012700  103736                   MOV     #STCDT,R0       ;SET UP THE DESTINATION ADDRESS.
103654  176010                   2$:     STCDF   AC0,(R0)        ;TEST INSTRUCTION.

103656  170204                           STFPS   R4              ;GET THE FPS.
103660  170305                           STST    R5              ;GET THE FEC.
103662  010102                           MOV     R1,R2           ;CHECK THE RESULT.
103664  062702  000010                   ADD     #10,R2
103670  012703  103736                   MOV     #STCDT,R3
103674  012700  000004                   MOV     #4,R0
103700  022223                   3$:     CMP     (R2)+,(R3)+
103702  001014                           BNE     10$             ;BRANCH IF INCORRECT.
103704  077003                           SOB     R0,3$

103706  016102  000032                   MOV     32(R1),R2
103712  020204                           CMP     R2,R4           ;IS THE FPS CORRECT?
103714  001007                           BNE     10$             ;BRANCH IF FPS INCORRECT.
```

```
103716  005702                          TST     R2              ;IF EXPECTED FPS IS NEGATIVE, THEN
103720  100003                          BPL     4$              ;GO AHEAD AND CHECK THE FEC.
103722  026105  000034                  CMP     34(R1),R5
103726  001002                          BNE     10$             ;BRANCH IF FEC IS INCORRECT.
103730  000161  000040          4$:     JMP     40(R1)          ;RETURN.
103734                          10$:
103734  104000                          EMT                     ;
103736  177777  177777  177777  STCDT:  -1,-1,-1,-1
103744  177777
103746                          YYYDONE:
103746  004767  014522                  JSR     PC,.RSET        ;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?).
                                ;*****************************************************************************
                                ;TEST 513       STCFD WITH ILLEGAL ACCUMULATOR TEST
                                ;*****************************************************************************
103752                          TS513:

103752  012700  040000                  MOV     #40000,R0       ;DISSABLE INTERRUPTS.
103756  170100                          LDFPS   R0
103760  176006                  ZZZ2:   STCFD   AC0,AC6         ;THIS TEST INSTRUCTION SHOULD CAUSE AN ERROR.

103762  170204                          STFPS   R4              ;GET FPS.
103764  170305                          STST    R5              ;GET FEC.
103766  020427  140000                  CMP     R4,#140000      ;IS FPS CORRECT?
103772  001004                          BNE     ZZZ10           ;BRANCH IF INCORRECT FPS.
103774  022705  000002                  CMP     #2,R5           ;IS FEC CORRECT?
104000  001001                          BNE     ZZZ10           ;BRANCH IF INCORRECT.
104002  000401                          BR      ZZZDONE

104004                          ZZZ10:
104004  104000                          EMT                     ;

104006                          ZZZDONE:
104006  004767  014462                  JSR     PC,.RSET        ;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?).

                                ;*****************************************************************************
                                ;TEST 514       CLRD TEST
                                ;*****************************************************************************
104012                          TS514:
104012  012700  104116                  MOV     #AABTP1,R0      ;SET UP OUTPUT BUFFER
104016  012701  104106                  MOV     #AABBF0,R1
104022  012702  000004                  MOV     #4,R2
104026  012021                  1$:     MOV     (R0)+,(R1)+
104030  077202                          SOB     R2,1$
104032  012700  104106                  MOV     #AABBF0,R0      ;SET UP DESTINATION OPERAND ADDRESS.
104036  012701  000213                  MOV     #213,R1         ;SET UP FPS.
104042  170101                          LDFPS   R1
104044  170410                  2$:     CLRD    (R0)            ;TEST INSTRUCTION.
```

```
104046  170205                  STFPS   R5              ;GET FPS.
104050  012702  000004          MOV     #4,R2           ;SEE IF RESULT CLEAR, 0.
104054  012701  104106          MOV     #AABBFO,R1
104060  005721          3$:     TST     (R1)+
104062  001010                  BNE     AAB2            ;BRANCH IF RESULT INCORRECT, NOT 0.
104064  077203                  SOB     R2,3$
104066  022705  000204          CMP     #204,R5         ;SEE IF FPS IS CORRECT.
104072  001004                  BNE     AAB2            ;BRANCH IF INCORRECT.
104074  020027  104106          CMP     RO,#AABBFO      ;SEE IF RO IS CORRECT.
104100  001001                  BNE     AAB2            ;BRANCH IF RO IS INCORRECT.
104102  000411                  BR      AABDONE


104104                  AAB2:
104104  104000                  EMT                     ;

                        ;THIS IS THE TEST DATA BUFFER, OUTPUT DATA BUFFER.
104106  073475          AABBFO: 73475
104110  067707                  67707
104112  127347                  127347
104114  056770                  56770
                        ;THIS IS THE DATA USED TO SET UP THE OUTPUT BUFFER.
104116  073475          AABTP1: 73475
104120  067707                  67707
104122  127347                  127347
104124  056770                  56770
104126                  AABDONE:
104126  004767  014342          JSR     PC,.RSET        ;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?).

                        ;*******************************************************************************
                        ;TEST 515       CLRD WITH ILLEGAL ACCUMULATOR TEST
                        ;*******************************************************************************
104132                  TS515:
104132  012700  040200          MOV     #40200,RO       ;SET UP THE FPS, NO INTERRUPTS AND FD=1.
104136  170100                  LDFPS   RO
104140  170407          CCB2:   CLRD    AC7             ;TEST INSTRUCTION.

104142  170204                  STFPS   R4              ;GET FPS.
104144  170305                  STST    R5              ;GET FEC.
104146  020427  140200          CMP     R4,#140200              ;IS THE FPS CORRECT?
104152  001004                  BNE     CCB10           ;BRANCH IF FPS IS INCORRECT.
104154  022705  000002          CMP     #2,R5           ;IS THE FEC CORRECT?
104160  001001                  BNE     CCB10           ;BRANCH IF FEC IS INCORRECT.
104162  000401                  BR      CCBDONE

104164                  CCB10:
104164  104000                  EMT                     ;
104166                  CCBDONE:
104166  004767  014302          JSR     PC,.RSET        ;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?).

                                  ;*******************************************************************************
                                  ;TEST 516        NEGF, ABSF AND TSTF SOURCE MODE 0 WITH ILLEGAL AC7, TEST
                                  ;*******************************************************************************
104172                            TS516:

104172  012700  040200                    MOV     #40200,R0                    ;SET UP THE FPS, FID=1 AND FD=1.
104176  170100                            LDFPS   R0
104200  170707                    VVB2:   NEGD    AC7               ;TEST INSTRUCTION.

104202  170204                            STFPS   R4                ;GET FPS.
104204  170305                            STST    R5                ;GET FEC.

104206  022704  140200                    CMP     #140200,R4                 ;IS FPS CORRECT?
104212  001004                            BNE     VVB10             ;BRANCH IF FPS IS INCORRECT.
104214  022705  000002                    CMP     #2,R5             ;IS FEC CORRECT?
104220  001001                            BNE     VVB10             ;BRANCH IF FEC IS INCORRECT.
104222  000401                            BR      VVBDONE
104224                            VVB10:
104224  104000                            EMT                       ;

104226                            VVBDONE:
104226  004767  014242                    JSR     PC,.RSET          ;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?).

                                  ;*******************************************************************************
                                  ;TEST 517        NEGF, ABSF AND TSTF SOURCE MODE 0 TEST
                                  ;*******************************************************************************
104232                            TS517:

104232  012700  000200                    MOV     #200,R0           ;SET FD MODE.
104236  170100                            LDFPS   R0
104240  012700  104332                    MOV     #DDBTP1,R0        ;SET UP AC0.
104244  172410                            LDD     (R0),AC0                   ;SET AC0 = 0
104246  005000                            CLR     R0                ;CLEAR THE FPS.
104250  170100                            LDFPS   R0
104252  012700  104342                    MOV     #DDBTP2,R0                 ;LOAD AC0 TO BE A FLOATING 0.
104256  172410                            LDF     (R0),AC0          ;SET AC0=ZERO
                                                                    ;FLOAT
104260  012700  000201                    MOV     #201,R0           ;SET FD MODE.
104264  170100                            LDFPS   R0
104266  170700                    DDB2:   NEGD    AC0               ;TEST INSTRUCTION

104270  170205                            STFPS   R5                ;GET FPS.
104272  012700  000200                    MOV     #200,R0           ;SET FD MODE.
104276  170100                            LDFPS   R0
104300  012700  104352                    MOV     #DDBBF0,R0        ;GET THE RESULT OUT OF AC0.
104304  174010                            STD     AC0,(R0)
                                                                    ;SEE IF THE RESULT IS CORRECT.
104306  012701  000004                    MOV     #4,R1
104312  005720                    1$:     TST     (R0)+
104314  001005                            BNE     DDB5              ;BRANCH IF THE RESULT IS INCORRECT.
```

```
104316  077103                          SOB     R1,1$
104320  022705  000204                  CMP     #204,R5         ;IS THE FPS CORRECT?
104324  001001                          BNE     DDB5            ;BRANCH IF THE FPS IS INCORRECT.
104326  000415                          BR      DDBDONE
104330                          DDB5:
104330  104000                          EMT                     ;

                                ;THESE ARE TEST DATA TABLES AND AN OUTPUT BUFFER.
104332  101112                  DDBTP1: 101112
104334  131415                          131415
104336  161710                          161710
104340  111213                          111213
104342  000000                  DDBTP2: 0
104344  000000                          0
104346  000000                          0
104350  000000                          0

104352  177777                  DDBBF0: -1
104354  177777                          -1
104356  177777                          -1
104360  177777                          -1

104362                          DDBDONE:
104362  004767  014106                  JSR     PC,.RSET        ;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?).

                                ;*************************************************************************
                                ;TEST 520       NEGF, ABSF AND TSTF SOURCE MODE 1 TEST
                                ;*************************************************************************
104366                          TS520:

104366  012700  104466                  MOV     #EEBTP1,R0      ;SET UP THE DATA BUFFER.
104372  012701  104506                  MOV     #EEBBF1,R1
104376  012702  000004                  MOV     #4,R2
104402  012021                  1$:     MOV     (R0)+,(R1)+
104404  077202                          SOB     R2,1$
104406  012700  000200                  MOV     #200,R0         ;SET FD MODE.
104412  170100                          LDFPS   R0
104414  012700  104506                  MOV     #EEBBF1,R0      ;SET UP THE OPERAND ADDRESS.
104420  012737  104516  000004          MOV     #EEB10,@#ERRVECT ;SET UP VECTOR 4 IN CASE OF ERROR.
104426  170710                  EEB2:   NEGD    (R0)            ;TEST INSTRUCTION.

104430  170205                          STFPS   R5              ;GET FPS.
104432  012701  104506                  MOV     #EEBBF1,R1      ;SEE IF RESULT IS CORRECT.
104436  012702  000004                  MOV     #4,R2
104442  005721                  1$:     TST     (R1)+
104444  001024                          BNE     EEB10           ;BRANCH IF NOT CORRECT.
104446  077203                          SOB     R2,1$

104450  020027  104506                  CMP     R0,#EEBBF1      ;IS R0 CORRECT?
104454  001020                          BNE     EEB10                   ;BRANCH IF NOT CORRECT.
104456  022705  000204                  CMP     #204,R5         ;IS THE FPS CORRECT?
104462  001015                          BNE     EEB10           ;BRANCH IF NOT CORRECT.
```

```
104464  000415                          BR      EEBDONE

                                ;THESE ARE TEST DATA TABLES AND A BUFFER.
104466  000177                  EEBTP1: 177
104470  167574                          167574
104472  137271                          137271
104474  107675                          107675
104476  177777                  EEBBF0: -1
104500  177777                          -1
104502  177777                          -1
104504  177777                          -1
104506  177777                  EEBBF1: -1
104510  177777                          -1
104512  177777                          -1
104514  177777                          -1
104516                          EEB10:
104516  104000                          EMT                             ;
104520                          EEBDONE:
104520  004767  013750                  JSR     PC,.RSET                ;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?).

                                ;*****************************************************************************
                                ;TEST 521       NEGF, ABSF AND TSTF SOURCE MODE 2 TEST
                                ;*****************************************************************************
104524                          TS521:

104524  012700  104624                  MOV     #FFBTP1,R0                      ;SET UP THE DATA BUFFER.
104530  012701  104634                  MOV     #FFBBF1,R1
104534  012702  000004                  MOV     #4,R2
104540  012021                  1$:     MOV     (R0)+,(R1)+
104542  077202                          SOB     R2,1$
104544  012700  000200                  MOV     #200,R0                 ;SET FD.
104550  170100                          LDFPS   R0
104552  012700  104634                  MOV     #FFBBF1,R0              ;SET UP THE OPERAND ADDRESS.
104556  012737  104644  000004          MOV     #FFB10,@#ERRVECT        ;SET UP VECTOR 4 IN CASE OF AN ERROR.

104564  170620                  FFB2:   ABSD    (R0)+                   ;TEST INSTRUCTION.

104566  170205                          STFPS   R5                      ;GET FPS.
104570  012701  104634                  MOV     #FFBBF1,R1                      ;CHECK RESULT.
104574  012702  000004                  MOV     #4,R2
104600  005721                  1$:     TST     (R1)+
104602  001020                          BNE     FFB10                   ;BRANCH IF INCORRECT.
104604  077203                          SOB     R2,1$

104606  020027  104644                  CMP     R0,#FFBBF1+10           ;IS R0 CORRECT?
104612  001014                          BNE     FFB10                   ;BRANCH IF INCORRECT.
104614  022705  000204                  CMP     #204,R5                 ;IS THE FPS CORRECT?
104620  001011                          BNE     FFB10                   ;BRANCH IF INCORRECT.
104622  000411                          BR      FFBDONE

                                ;THESE ARE TEST DATA TABLES AND DATA BUFFER.
104624  000177                  FFBTP1: 177
```

```
104626  167574                                  167574
104630  137271                                  137271
104632  107675                                  107675
104634  177777                          FFBBF1: -1
104636  177777                                  -1
104640  177777                                  -1
104642  177777                                  -1
104644                                  FFB10:
104644  104000                                  EMT                     ;
104646                                  FFBDONE:
104646  004767  013622                          JSR     PC,.RSET        ;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?).
                                        ;*******************************************************************************
                                        ;TEST 522       NEGF, ABSF AND TSTF SOURCE MODE 4 TEST
                                        ;*******************************************************************************
104652                                  TS522:

104652  012700  104752                          MOV     #GGBTP1,R0              ;SET UP THE DATA BUFFER.
104656  012701  104762                          MOV     #GGBBF0,R1
104662  012702  000004                          MOV     #4,R2
104666  012021                          1$:     MOV     (R0)+,(R1)+
104670  077202                                  SOB     R2,1$
104672  012700  000200                          MOV     #200,R0         ;SET FD.
104676  170100                                  LDFPS   R0
104700  012700  104772                          MOV     #GGBBF1,R0      ;SET UP THE OPERAND ADDRESS.
104704  012737  105002  000004                  MOV     #GGB10,@#ERRVECT ;SET UP VECTOR 4 IN CASE OF AN ERROR.

104712  170640                          GGB2:   ABSD    -(R0)           ;TEST INSTRUCTION.

104714  170205                                  STFPS   R5              ;GET FPS.
104716  012701  104762                          MOV     #GGBBF0,R1              ;CHECK RESULT.
104722  012702  000004                          MOV     #4,R2
104726  005721                          1$:     TST     (R1)+
104730  001024                                  BNE     GGB10           ;BRANCH IF INCORRECT.
104732  077203                                  SOB     R2,1$

104734  020027  104762                          CMP     R0,#GGBBF0      ;IS R0 CORRECT?
104740  001020                                  BNE     GGB10           ;BRANCH IF INCORRECT.
104742  022705  000204                          CMP     #204,R5         ;IS THE FPS CORRECT?
104746  001015                                  BNE     GGB10           ;BRANCH IF INCORRECT.
104750  000415                                  BR      GGBDONE

                                        ;THESE ARE TEST DATA TABLES AND DATA BUFFER.
104752  000177                          GGBTP1: 177
104754  117273                                  117273
104756  147576                                  147576
104760  177071                                  177071
104762  177777                          GGBBF0: -1
104764  177777                                  -1
104766  177777                                  -1
104770  177777                                  -1
104772  177777                          GGBBF1: -1
104774  177777                                  -1
```

```
104776 177777                          -1
105000 177777                          -1
105002                          GGB10:
105002 104000                          EMT                        ;
105004                          GGBDONE:
105004 004767 013464                   JSR     PC,.RSET           ;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?).
                                ;*******************************************************************************
                                ;TEST 523        NEGF, ABSF AND TSTF SOURCE MODE 3 TEST
                                ;*******************************************************************************
105010                          TS523:

105010 012700 105110                   MOV     #HHBTP1,R0                   ;SET UP THE DATA BUFFER.
105014 012701 105130                   MOV     #HHBBF0,R1
105020 012702 000010                   MOV     #10,R2
105024 012021                   1$:    MOV     (R0)+,(R1)+
105026 077202                          SOB     R2,1$
105030 012700 000200                   MOV     #200,R0            ;SET FD.
105034 170100                          LDFPS   R0
105036 012700 105140                   MOV     #HHBBF1,R0         ;SET UP THE OPERAND ADDRESS.
105042 012737 105150 000004            MOV     #HHB10,@#ERRVECT ;SET UP VECTOR 4 IN CASE OF AN ERROR.

105050 170630                   HHB2:  ABSD    @(R0)+             ;TEST INSTRUCTION.

105052 170205                          STFPS   R5                 ;GET FPS.
105054 012701 105130                   MOV     #HHBBF0,R1                   ;CHECK RESULT.
105060 012702 000004                   MOV     #4,R2
105064 005721                   1$:    TST     (R1)+
105066 001030                          BNE     HHB10              ;BRANCH IF INCORRECT.
105070 077203                          SOB     R2,1$
105072 020027 105142                   CMP     R0,#HHBBF1+2       ;IS R0 CORRECT?
105076 001024                          BNE     HHB10              ;BRANCH IF INCORRECT.
105100 022705 000204                   CMP     #204,R5            ;IS THE FPS CORRECT?
105104 001021                          BNE     HHB10              ;BRANCH IF INCORRECT.
105106 000421                          BR      HHBDONE

                                ;THESE ARE TEST DATA TABLES AND DATA BUFFER.
105110 000177                   HHBTP1: 177
105112 147576                          147576
105114 177071                          177071
105116 107576 105130 177777            107576,HHBBF0,-1,-1,-1
105124 177777 177777
105130 177777                   HHBBF0: -1
105132 177777                          -1
105134 177777                          -1
105136 177777                          -1
105140 177777                   HHBBF1: -1
105142 177777                          -1
105144 177777                          -1
105146 177777                          -1
105150                          HHB10:
105150 104000                          EMT                        ;
105152                          HHBDONE:
```

```
105152  004767  013316                  JSR     PC,.RSET                ;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?).
                                ;*******************************************************************************
                                ;TEST 524       NEGF, ABSF AND TSTF SOURCE MODE 5 TEST
                                ;*******************************************************************************
105156                          TS524:
105156  012700  105256                  MOV     #IIBTP1,R0                      ;SET UP THE DATA BUFFER.
105162  012701  105276                  MOV     #IIBBF0,R1
105166  012702  000010                  MOV     #10,R2
105172  012021                  1$:     MOV     (R0)+,(R1)+
105174  077202                          SOB     R2,1$
105176  012700  000200                  MOV     #200,R0                 ;SET FD.
105202  170100                          LDFPS   R0
105204  012700  105310                  MOV     #IIBBF1+2,R0            ;SET UP THE OPERAND ADDRESS.
105210  012737  105316  000004          MOV     #IIB10,@#ERRVECT        ;SET UP VECTOR 4 IN CASE OF AN ERROR.

105216  170750                  IIB2:   NEGD    @-(R0)                  ;TEST INSTRUCTION.

105220  170205                          STFPS   R5                      ;GET FPS.
105222  012701  105276                  MOV     #IIBBF0,R1                      ;CHECK RESULT.
105226  012702  000004                  MOV     #4,R2
105232  005721                  1$:     TST     (R1)+
105234  001030                          BNE     IIB10                   ;BRANCH IF INCORRECT.
105236  077203                          SOB     R2,1$
105240  020027  105306                  CMP     R0,#IIBBF1              ;IS R0 CORRECT?
105244  001024                          BNE     IIB10                   ;BRANCH IF INCORRECT.
105246  022705  000204                  CMP     #204,R5                 ;IS THE FPS CORRECT?
105252  001021                          BNE     IIB10                   ;BRANCH IF INCORRECT.
105254  000421                          BR      IIBDONE

                                ;THESE ARE TEST DATA TABLES AND DATA BUFFER.
105256  000176                  IIBTP1: 176
105260  177074                          177074
105262  127374                          127374
105264  157677  105276  177777          157677,IIBBF0,-1,-1,-1
105272  177777  177777
105276  177777                  IIBBF0: -1
105300  177777                          -1
105302  177777                          -1
105304  177777                          -1
105306  177777                  IIBBF1: -1
105310  177777                          -1
105312  177777                          -1
105314  177777                          -1

105316                          IIB10:
105316  104000                          EMT                             ;
105320                          IIBDONE:
105320  004767  013150                  JSR     PC,.RSET                ;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?).
                                        ;*******************************************************************************
                                        ;TEST 525       NEGF, ABSF AND TSTF SOURCE MODE 6 TEST
                                        ;*******************************************************************************
105324                                  TS525:

105324  012700  105426                          MOV     #JJBTP1,R0              ;SET UP THE DATA BUFFER.
105330  012701  105440                          MOV     #JJBBF0,R1
105334  012702  000004                          MOV     #4,R2
105340  012021                          1$:     MOV     (R0)+,(R1)+
105342  077202                                  SOB     R2,1$
105344  012700  000200                          MOV     #200,R0         ;SET FD.
105350  170100                                  LDFPS   R0
105352  012700  105431                          MOV     #JJBBF0-7,R0    ;SET UP THE OPERAND ADDRESS.
105356  012737  105450  000004                  MOV     #JJB10,@#ERRVECT ;SET UP VECTOR 4 IN CASE OF AN ERROR.

105364  170660  000007                  JJB2:   ABSD    7(R0)           ;TEST INSTRUCTION.

105370  170205                                  STFPS   R5              ;GET FPS.
105372  012701  105440                          MOV     #JJBBF0,R1              ;CHECK RESULT.
105376  012702  000004                          MOV     #4,R2
105402  005721                          1$:     TST     (R1)+
105404  001021                                  BNE     JJB10           ;BRANCH IF INCORRECT.
105406  077203                                  SOB     R2,1$
105410  020027  105431                          CMP     R0,#JJBBF0-7    ;IS R0 CORRECT?
105414  001015                                  BNE     JJB10           ;BRANCH IF INCORRECT.
105416  022705  000204                          CMP     #204,R5         ;IS THE FPS CORRECT?
105422  001012                                  BNE     JJB10           ;BRANCH IF INCORRECT.
105424  000412                                  BR      JJBDONE

                                        ;THESE ARE TEST DATA TABLES AND DATA BUFFER.
105426  000177                          JJBTP1: 177
105430  161524                                  161524
105432  131273                                  131273
105434  107174  000000                          107174,
105440  177777                          JJBBF0: -1
105442  177777                                  -1
105444  177777                                  -1
105446  177777                                  -1
105450                                  JJB10:
105450  104000                                  EMT                     ;
105452                                  JJBDONE:
105452  004767  013016                          JSR     PC,.RSET        ;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?).
                                        ;*******************************************************************************
                                        ;TEST 526       NEGF, ABSF AND TSTF SOURCE MODE 7 TEST
                                        ;*******************************************************************************
105456                                  TS526:

105456  012700  105560                          MOV     #KKBTP1,R0              ;SET UP THE DATA BUFFER.
105462  012701  105600                          MOV     #KKBBF0,R1
105466  012702  000010                          MOV     #10,R2
105472  012021                          1$:     MOV     (R0)+,(R1)+
```

```
105474  077202                               SOB     R2,1$
105476  012700  000200                       MOV     #200,R0           ;SET FD.
105502  170100                               LDFPS   R0
105504  012700  105601                       MOV     #KKBBF1-7,R0      ;SET UP THE OPERAND ADDRESS.
105510  012737  105620  000004               MOV     #KKB10,@#ERRVECT  ;SET UP VECTOR 4 IN CASE OF AN ERROR.

105516  170770  000007               KKB2:   NEGD    @7(R0)            ;TEST INSTRUCTION.

105522  170205                               STFPS   R5                ;GET FPS.
105524  012701  105600                       MOV     #KKBBF0,R1                  ;CHECK RESULT.
105530  012702  000004                       MOV     #4,R2
105534  005721                       1$:     TST     (R1)+
105536  001030                               BNE     KKB10             ;BRANCH IF INCORRECT.
105540  077203                               SOB     R2,1$
105542  020027  105601                       CMP     R0,#KKBBF1-7      ;IS R0 CORRECT?
105546  001024                               BNE     KKB10             ;BRANCH IF INCORRECT.
105550  022705  000204                       CMP     #204,R5           ;IS THE FPS CORRECT?
105554  001021                               BNE     KKB10             ;BRANCH IF INCORRECT
105556  000421                               BR      KKBDONE

                                     ;THESE ARE TEST DATA TABLES AND DATA BUFFER.
105560  000177                       KKBTP1: 177
105562  167574                               167574
105564  137271                               137271
105566  107675  105600  177777               107675,KKBBF0,-1,-1,-1
105574  177777  177777
105600  177777                       KKBBF0: -1
105602  177777                               -1
105604  177777                               -1
105606  177777                               -1
105610  177777                       KKBBF1: -1
105612  177777                               -1
105614  177777                               -1
105616  177777                               -1
105620                               KKB10:
105620  104000                               EMT                       ;
105622                               KKBDONE:
105622  004767  012646                       JSR     PC,.RSET          ;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?).
                                     ;*************************************************************************
                                     ;TEST 527        NEGF, ABSF AND TSTF SOURCE MODE 6, GR7, TEST
                                     ;*************************************************************************
105626                               TS527:
105626  012700  105716                       MOV     #LLBTP1,R0                  ;SET UP THE DATA BUFFER.
105632  012701  105726                       MOV     #LLBBF0,R1
105636  012702  000004                       MOV     #4,R2
105642  012021                       1$:     MOV     (R0)+,(R1)+
105644  077202                               SOB     R2,1$
105646  012700  000200                       MOV     #200,R0           ;SET FD.
105652  170100                               LDFPS   R0
105654  012737  105736  000004               MOV     #LLB10,@#ERRVECT  ;SET UP VECTOR 4 IN CASE OF AN ERROR.

105662  170767  000040               LLB2:   NEGD    LLBBF0            ;TEST INSTRUCTION.
```

```
105666  170205                          STFPS   R5              ;GET FPS.
105670  012701  105726                  MOV     #LLBBF0,R1              ;CHECK RESULT.
105674  012702  000004                  MOV     #4,R2
105700  005721                  1$:     TST     (R1)+
105702  001015                          BNE     LLB10           ;BRANCH IF INCORRECT.
105704  077203                          SOB     R2,1$
105706  022705  000204                  CMP     #204,R5         ;IS THE FPS CORRECT?
105712  001011                          BNE     LLB10           ;BRANCH IF INCORRECT.
105714  000411                          BR      LLBDONE

                                ;THESE ARE TEST DATA TABLES AND DATA BUFFER.
105716  000127                  LLBTP1: 127
105720  137475                          137475
105722  147372                          147372
105724  117057                          117057
105726  177777                  LLBBF0: -1
105730  177777                          -1
105732  177777                          -1
105734  177777                          -1

105736                          LLB10:
105736  104000                          EMT                     ;
105740                          LLBDONE:
105740  004767  012530                  JSR     PC,.RSET        ;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?).
                                ;*******************************************************************************
                                ;TEST 530       NEGF, ABSF AND TSTF SOURCE MODE 7, GR7, TEST
                                ;*******************************************************************************
105744                          TS530:

105744  012700  106034                  MOV     #MMBTP1,R0              ;SET UP THE DATA BUFFER.
105750  012701  106054                  MOV     #MMBBF0,R1
105754  012702  000010                  MOV     #10,R2
105760  012021                  1$:     MOV     (R0)+,(R1)+
105762  077202                          SOB     R2,1$
105764  012700  000200                  MOV     #200,R0         ;SET FD.
105770  170100                          LDFPS   R0
105772  012737  106074  000004          MOV     #MMB10,@#ERRVECT ;SET UP VECTOR 4 IN CASE OF AN ERROR.

106000  170677  000060          MMB2:   ABSD    @MMBBF1         ;TEST INSTRUCTION.

106004  170205                          STFPS   R5              ;GET FPS.
106006  012701  106054                  MOV     #MMBBF0,R1              ;CHECK RESULT.
106012  012702  000004                  MOV     #4,R2
106016  005721                  1$:     TST     (R1)+
106020  001025                          BNE     MMB10           ;BRANCH IF INCORRECT.
106022  077203                          SOB     R2,1$
106024  022705  000204                  CMP     #204,R5         ;IS THE FPS CORRECT?
106030  001021                          BNE     MMB10           ;BRANCH IF INCORRECT.
106032  000421                          BR      MMBDONE

                                ;THESE ARE TEST DATA TABLES AND DATA BUFFER.
```

```
106034  000137                          MMBTP1: 137
106036  045607                                  045607
106040  101230                                  101230
106042  045607  106054  177777                  45607,MMBBF0,-1,-1,-1
106050  177777  177777
106054  177777                          MMBBF0: -1
106056  177777                                  -1
106060  177777                                  -1
106062  177777                                  -1
106064  177777                          MMBBF1: -1
106066  177777                                  -1
106070  177777                                  -1
106072  177777                                  -1

106074                                  MMB10:
106074  104000                                  EMT                             ;
106076                                  MMBDONE:
106076  004767  012372                          JSR     PC,.RSET                ;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?).
                                        ;*************************************************************************
                                        ;TEST 531       SPECIAL DEST, MODE 0, TEST
                                        ;*************************************************************************
106102                                  TS531:

106102  012700  000200                          MOV     #200,R0                 ;SET FD.
106106  170100                                  LDFPS   R0
106110  012700  106170                          MOV     #NNBTP1,R0                      ;SET UP AC0.
106114  172410                                  LDD     (R0),AC0
106116  170700                          NNB2:   NEGD    AC0                     ;TEST INSTRUCTION.

106120  170205                                  STFPS   R5                      ;GET FPS.
106122  012700  000200                          MOV     #200,R0                 ;SET FD.
106126  170100                                  LDFPS   R0
106130  012700  106210                          MOV     #NNBBF0,R0                      ;GET THE RESULT.
106134  174010                                  STD     AC0,(R0)
106136  012700  106210                          MOV     #NNBBF0,R0                      ;IS THE RESULT CORRECT?
106142  012701  106200                          MOV     #NNBTP2,R1
106146  012702  000004                          MOV     #4,R2
106152  022021                          1$:     CMP     (R0)+,(R1)+
106154  001021                                  BNE     NNB10                   ;BRANCH IF INCORRECT.
106156  077203                                  SOB     R2,1$
106160  022705  000210                          CMP     #210,R5                 ;IS THE FPS CORRECT?
106164  001015                                  BNE     NNB10                   ,BRANCH IF INCORRECT.
106166  000415                                  BR      NNBDONE

                                        ;THESE ARE DATA TABLES AND A DATA BUFFER.
106170  013572                          NNBTP1: 013572
106172  046013                                  46013
106174  057246                                  57246
106176  013570                                  013570
106200  113572                          NNBTP2: 113572
106202  046013                                  46013
106204  057246                                  57246
```

```
106206  013570                          013570
106210  000000                  NNBBF0: 0
106212  000000                          0
106214  000000                          0
106216  000000                          0

106220                          NNB10:
106220  104000                          EMT                             ;
106222                          NNBDONE:
106222  004767  012246                  JSR     PC,.RSET                ;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?).
                                ;*************************************************************************
                                ;TEST 532       SPECIAL DEST, MODE 1, TEST
                                ;*************************************************************************
106226                          TS532:

106226  012701  106330                  MOV     #OOBTP1,R1              ;SET UP THE DATA BUFFER.
106232  012700  106340                  MOV     #OOBTP2,R0
106236  012702  000004                  MOV     #4,R2
106242  012021                  1$:     MOV     (R0)+,(R1)+
106244  077202                          SOB     R2,1$
106246  012700  106330                  MOV     #OOBTP1,R0
106252  042710  100000                  BIC     #100000,(R0)            ;MAKE OPERAND POSITIVE.
106256  012701  000200                  MOV     #200,R1                 ;SET FD.
106262  170101                          LDFPS   R1

106264  170710                  OOB2:   NEGD    (R0)                    ;TEST INSTRUCTION.
106266  170205                          STFPS   R5                      ;GET FPS.
106270  012701  106330                  MOV     #OOBTP1,R1                      ;IS THE RESULT CORRECT.
106274  012702  106340                  MOV     #OOBTP2,R2
106300  012703  000004                  MOV     #4,R3
106304  022122                  1$:     CMP     (R1)+,(R2)+
106306  001020                          BNE     OOB10                   ;BRANCH IF INCORRECT.
106310  077303                          SOB     R3,1$
106312  022700  106330                  CMP     #OOBTP1,R0              ;IS R0 CORRECT.
106316  001014                          BNE     OOB10                   ;BRANCH IF INCORRECT.
106320  022705  000210                  CMP     #210,R5                 ;IS THE FPS CORRECT?
106324  001011                          BNE     OOB10                   ;BRANCH IF INCORRECT.
106326  000411                          BR      OOBDONE

                                ;THESE ARE DATA TABLES AND A DATA BUFFER.
106330  023245                  OOBTP1: 023245
106332  026720                          26720
106334  122324                          122324
106336  052672                          52672
106340  123245                  OOBTP2: 123245
106342  026720                          26720
106344  122324                          122324
106346  052672                          52672

106350                          OOB10:
106350  104000                          EMT                             ;
106352                          OOBDONE:
```

```
106352  004767  012116          JSR     PC,.RSET        ;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?).
                                ;*************************************************************
                                ;TEST 533       SPECIAL DEST, MODE 2, TEST
                                ;*************************************************************
106356                          TS533:

106356  012701  106460          MOV     #PPBTP1,R1      ;SET UP THE DATA BUFFER.
106362  012700  106470          MOV     #PPBTP2,R0
106366  012702  000004          MOV     #4,R2
106372  012021                  1$:     MOV     (R0)+,(R1)+
106374  077202                  SOB     R2,1$
106376  012700  106460          MOV     #PPBTP1,R0
106402  042710  100000          BIC     #100000,(R0)    ;MAKE OPERAND POSITIVE.
106406  012701  000200          MOV     #200,R1         ;SET FD.
106412  170101                  LDFPS   R1

106414  170720                  PPB2:   NEGD    (R0)+           ;TEST INSTRUCTION.

106416  170205                  STFPS   R5              ;GET FPS.
106420  012701  106460          MOV     #PPBTP1,R1              ;IS THE RESULT CORRECT.
106424  012702  106470          MOV     #PPBTP2,R2
106430  012703  000004          MOV     #4,R3
106434  022122                  1$:     CMP     (R1)+,(R2)+
106436  001020                  BNE     PPB10           ;BRANCH IF INCORRECT.
106440  077303                  SOB     R3,1$
106442  022700  106470          CMP     #PPBTP1+10,R0   ;IS R0 CORRECT.
106446  001014                  BNE     PPB10           ;BRANCH IF INCORRECT.
106450  022705  000210          CMP     #210,R5         ;IS THE FPS CORRECT?
106454  001011                  BNE     PPB10           ;BRANCH IF INCORRECT.
106456  000411                  BR      PPBDONE

                                ;THESE ARE DATA TABLES AND A DATA BUFFER.
106460  023245                  PPBTP1: 023245
106462  026720                          26720
106464  122324                          122324
106466  052672                          52672
106470  123245                  PPBTP2: 123245
106472  026720                          26720
106474  122324                          122324
106476  052672                          52672

106500                          PPB10:
106500  104000                          EMT                     ;
106502                          PPBDONE:
106502  004767  011766                  JSR     PC,.RSET        ;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?).
                                ;*************************************************************
                                ;TEST 534       SPECIAL DEST, MODE 4, TEST
                                ;*************************************************************
```

```
106506                           TS534:
106506  012701  106612                   MOV     #QQBTP1,R1      ;SET UP THE DATA BUFFER.
106512  012700  106632                   MOV     #QQBTP2,R0
106516  012702  000004                   MOV     #4,R2
106522  012021                   1$:     MOV     (R0)+,(R1)+
106524  077202                           SOB     R2,1$
106526  012700  106622                   MOV     #QQBTP1+10,R0
106532  042760  100000  177770           BIC     #100000,-10(R0) ;MAKE OPERAND POSITIVE.
106540  012701  000200                   MOV     #200,R1         ;SET FD.
106544  170101                           LDFPS   R1

106546  170740                   QQB2:   NEGD    -(R0)           ;TEST INSTRUCTION.

106550  170205                           STFPS   R5              ;GET FPS.
106552  012701  106612                   MOV     #QQBTP1,R1              ;IS THE RESULT CORRECT.
106556  012702  106632                   MOV     #QQBTP2,R2
106562  012703  000004                   MOV     #4,R3
106566  022122                   1$:     CMP     (R1)+,(R2)+
106570  001024                           BNE     QQB10           ;BRANCH IF INCORRECT.
106572  077303                           SOB     R3,1$
106574  022700  106612                   CMP     #QQBTP1,R0      ;IS R0 CORRECT.
106600  001020                           BNE     QQB10           ;BRANCH IF INCORRECT.
106602  022705  000210                   CMP     #210,R5         ;IS THE FPS CORRECT?
106606  001015                           BNE     QQB10           ;BRANCH IF INCORRECT.
106610  000415                           BR      QQBDONE

                                 ;THESE ARE DATA TABLES AND A DATA BUFFER.
106612  023245                   QQBTP1: 023245
106614  026720                           26720
106616  122324                           122324
106620  052672                           52672
106622  177777  177777  177777           .WORD   -1,-1,-1,-1
106630  177777
106632  123245                   QQBTP2: 123245
106634  026720                           26720
106636  122324                           122324
106640  052672                           52672

106642                           QQB10:
106642  104000                           EMT                     ;
106644                           QQBDONE:
106644  004767  011624                   JSR     PC,.RSET        ;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?).

                                 ;*************************************************************************
                                 ;TEST 535        SPECIAL DEST. MODE 3. TEST
                                 ;*************************************************************************
106650                           TS535:

106650  012701  106760                   MOV     #RRBTP1,R1      ;SET UP THE DATA BUFFER.
106654  012700  106770                   MOV     #RRBTP2,R0
106660  012702  000004                   MOV     #4,R2
106664  012021                   1$:     MOV     (R0)+,(R1)+
```

```
106666  077202                          SOB     R2,1$
106670  012700  107000                  MOV     #RRBTP3,R0
106674  012710  106760                  MOV     #RRBTP1,(R0)
106700  042737  100000  106760          BIC     #100000,@#RRBTP1        ;MAKE THE OPERAND POSITIVE.
106706  012701  000200                  MOV     #200,R1         ;SET FD.
106712  170101                          LDFPS   R1

106714  170730                  RRB2:   NEGD    @(R0)+          ;TEST INSTRUCTION.

106716  170205                          STFPS   R5              ;GET FPS.
106720  012701  106760                  MOV     #RRBTP1,R1              ;IS THE RESULT CORRECT.
106724  012702  106770                  MOV     #RRBTP2,R2
106730  012703  000004                  MOV     #4,R3
106734  022122                  1$:     CMP     (R1)+,(R2)+
106736  001021                          BNE     RRB10           ;BRANCH IF INCORRECT.
106740  077303                          SOB     R3,1$
106742  022700  107002                  CMP     #RRBTP3+2,R0    ;IS R0 CORRECT.
106746  001015                          BNE     RRB10           ;BRANCH IF INCORRECT
106750  022705  000210                  CMP     #210,R5         ;IS THE FPS CORRECT?
106754  001012                          BNE     RRB10           ;BRANCH IF INCORRECT.
106756  000412                          BR      RRBDONE

                                ;THESE ARE DATA TABLES AND A DATA BUFFER.
106760  023245                  RRBTP1: 023245
106762  026720                          26720
106764  122324                          122324
106766  052672                          52672
106770  123245                  RRBTP2: 123245
106772  026720                          26720
106774  123324                          123324
106776  052672                          52672
107000  106760                  RRBTP3: RRBTP1

107002                          RRB10:
107002  104000                          EMT                     ;
107004                          RRBDONE:
107004  004767  011464                  JSR     PC,.RSET        ;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?).

                                ;*************************************************************************
                                ;TEST 536       SPECIAL DEST, MODE 5, TEST
                                ;*************************************************************************
107010                          TS536:
107010  012701  107122                  MOV     #SSBTP1,R1      ;SET UP THE DATA BUFFER.
107014  012700  107132                  MOV     #SSBTP2,R0
107020  012702  000004                  MOV     #4,R2
107024  012021                  1$:     MOV     (R0)+,(R1)+
107026  077202                          SOB     R2,1$
107030  012700  107144                  MOV     #SSBTP3+2,R0
107034  012760  107122  177776          MOV     #SSBTP1,-2(R0)
107042  042737  100000  107122          BIC     #100000,@#SSBTP1        ;MAKE THE OPERAND POSITIVE.
107050  012701  000200                  MOV     #200,R1         ;SET FD.
107054  170101                          LDFPS   R1
```

```

107056  170750                 SSB2:   NEGD    @-(R0)          ;TEST INSTRUCTION.

107060  170205                         STFPS   R5              ;GET FPS.
107062  012701  107122                 MOV     #SSBTP1,R1              ;IS THE RESULT CORRECT.
107066  012702  107132                 MOV     #SSBTP2,R2
107072  012703  000004                 MOV     #4,R3
107076  022122                 1$:     CMP     (R1)+,(R2)+
107100  001021                         BNE     SSB10           ;BRANCH IF INCORRECT.
107102  077303                         SOB     R3,1$
107104  022700  107142                 CMP     #SSBTP3,R0      ;IS R0 CORRECT.
107110  001015                         BNE     SSB10           ;BRANCH IF INCORRECT.
107112  022705  000210                 CMP     #210,R5         ;IS THE FPS CORRECT?
107116  001012                         BNE     SSB10           ;BRANCH IF INCORRECT.
107120  000412                         BR      SSBDONE

                               ;THESE ARE DATA TABLES AND A DATA BUFFER.
107122  023245                 SSBTP1: 023245
107124  026720                         26720
107126  122324                         122324
107130  052672                         52672
107132  123245                 SSBTP2: 123245
107134  026270                         26270
107136  122324                         122324
107140  052672                         52672
107142  107122                 SSBTP3: SSBTP1

107144                         SSB10:
107144  104000                         EMT                     ;
107146                         SSBDONE:
107146  004767  011322                 JSR     PC,.RSET        ;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?).
                               ;*****************************************************************************
                               ;TEST 537        SPECIAL DEST, FLOATING MODE 2, TEST
                               ;*****************************************************************************
107152                         TS537:
107152  012701  107254                 MOV     #TTBTP1,R1      ;SET UP THE DATA BUFFER.
107156  012700  107264                 MOV     #TTBTP2,R0
107162  012702  000004                 MOV     #4,R2
107166  012021                 1$:     MOV     (R0)+,(R1)+
107170  077202                         SOB     R2,1$
107172  012700  107254                 MOV     #TTBTP1,R0
107176  042710  100000                 BIC     #100000,(R0)    ;MAKE OPERAND POSITIVE.
107202  012701  000000                 MOV     #000,R1         ;SET FD.
107206  170101                         LDFPS   R1

107210  170720                 TTB2:   NEGF    (R0)+           ;TEST INSTRUCTION.

107212  170205                         STFPS   R5              ;GET FPS.
107214  012701  107254                 MOV     #TTBTP1,R1              ;IS THE RESULT CORRECT.
107220  012702  107264                 MOV     #TTBTP2,R2
107224  012703  000004                 MOV     #4,R3
107230  022122                 1$:     CMP     (R1)+,(R2)+
```

```
107232  001020                          BNE     TTB10           ;BRANCH IF INCORRECT.
107234  077303                          SOB     R3,1$
107236  022700  107260                  CMP     #TTBTP1+4,R0    ;IS R0 CORRECT.
107242  001014                          BNE     TTB10           ;BRANCH IF INCORRECT.
107244  022705  000010                  CMP     #010,R5         ;IS THE FPS CORRECT?
107250  001011                          BNE     TTB10           ;BRANCH IF INCORRECT.
107252  000411                          BR      TTBDONE

                                ;THESE ARE DATA TABLES AND A DATA BUFFER.
107254  023245                  TTBTP1: 023245
107256  026720                          26720
107260  122324                          122324
107262  052672                          52672
107264  123245                  TTBTP2: 123245
107266  026720                          26720
107270  122324                          122324
107272  052672                          52672

107274                          TTB10:
107274  104000                          EMT                     ;
107276                          TTBDONE:
107276  004767  011172                  JSR     PC,.RSET        ;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?).
                                ;*************************************************************************
                                ;TEST 540       SPECIAL DEST, MODE2, GR7 (IMMEDIATE), TEST
                                ;*************************************************************************
107302                          TS540:
107302  012700  107420                  MOV     #UUBTP2,R0
107306  012701  107346                  MOV     #UUBTP1,R1      ;SET UP THE DATA BUFFER.
107312  012702  000004                  MOV     #4,R2
107316  012021                  1$:     MOV     (R0)+,(R1)+
107320  077202                          SOB     R2,1$
107322  012700  107346                  MOV     #UUBTP1,R0
107326  042737  100000  107346          BIC     #100000,@#UUBTP1        ;MAKE THE OPERAND POSITIVE.
107334  012701  000200                  MOV     #200,R1         ;SET FD.
107340  170101                          LDFPS   R1
107342  005001                          CLR     R1

107344  170727                  UUB2:   NEGD    (R7)+           ;TEST INSTRUCTION.
107346  005201  005201  005201  UUBTP1: 5201,5201,5201,5201
107354  005201
                                ;NOTE THAT AFTER EXECUTING THIS INSTRUCTION R1 SHOULD CONTAIN 3.
107356  170205                          STFPS   R5              ;GET FPS.
107360  012703  107346                  MOV     #UUBTP1,R3      ;IS THE RESULT CORRECT.
107364  012702  107420                  MOV     #UUBTP2,R2
107370  012704  000004                  MOV     #4,R4
107374  022322                  1$:     CMP     (R3)+,(R2)+
107376  001014                          BNE     UUB10           ;BRANCH IF INCORRECT.
107400  077403                          SOB     R4,1$
107402  022701  000003                  CMP     #3,R1           ;WAS R1 INCREMENTED CORRECTLY.
107406  001010                          BNE     UUB10           ;BRANCH IF INCORRECT.
107410  022705  000210                  CMP     #210,R5         ;IS THE FPS CORRECT?
107414  001005                          BNE     UUB10           ;BRANCH IF INCORRECT.
```

```
107416  000405                          BR      UUBDONE

                                ;THESE ARE DATA TABLE.
107420  105201                  UUBTP2: 105201
107422  005201                          5201
107424  005201                          5201
107426  005201                          5201

107430                          UUB10:
107430  104000                          EMT                     ;
107432                          UUBDONE:
107432  004767  011036                  JSR     PC,.RSET        ;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?).
                                ;*********************************************************************
                                ;TEST 541       SPECIAL DEST, MODE 6, TEST
                                ;*********************************************************************
107436                          TS541:
107436  012701  107552                  MOV     #XXBTP1,R1      ;SET UP THE DATA BUFFER.
107442  012700  107562                  MOV     #XXBTP2,R0
107446  012702  000004                  MOV     #4,R2
107452  012021                  1$:     MOV     (R0)+,(R1)+
107454  077202                          SOB     R2,1$
107456  012700  102351                  MOV     #XXBTP1-5201,R0
107462  042737  100000  107552          BIC     #100000,@#XXBTP1;MAKE OPERAND POSITIVE.
107470  012701  000200                  MOV     #200,R1         ;SET FD.
107474  170101                          LDFPS   R1

107476  005001                          CLR     R1
107500  170760  005201          XXB2:   NEGD    5201(R0)                ;TEST INSTRUCTION.

107504  170205                          STFPS   R5              ;GET FPS.
107506  005701                          TST     R1
107510  001030                          BNE     XXB10           ;WAS THE PC CORRECT AFTER EXECUTION?
107512  012701  107552                  MOV     #XXBTP1,R1              ;IS THE RESULT CORRECT.
107516  012702  107562                  MOV     #XXBTP2,R2
107522  012703  000004                  MOV     #4,R3
107526  022122                  1$:     CMP     (R1)+,(R2)+
107530  001020                          BNE     XXB10           ;BRANCH IF INCORRECT.
107532  077303                          SOB     R3,1$
107534  022700  102351                  CMP     #XXBTP1-5201,R0 ;IS R0 CORRECT.
107540  001014                          BNE     XXB10           ;BRANCH IF INCORRECT.
107542  022705  000210                  CMP     #210,R5         ;IS THE FPS CORRECT?
107546  001011                          BNE     XXB10           ;BRANCH IF INCORRECT.
107550  000411                          BR      XXBDONE

                                ;THESE ARE DATA TABLES AND A DATA BUFFER.
107552  023245                  XXBTP1: 023245
107554  026720                          26720
107556  122324                          122324
107560  052672                          52672
107562  123245                  XXBTP2: 123245
107564  026720                          26720
107566  122324                          122324
```

```
107570  052672                                    52672


107572                          XXB10:
107572  104000                          EMT                             ;
107574                          XXBDONE:
107574  004767  010674                  JSR     PC,.RSET                ;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?).

                                ;*******************************************************************************
                                ;TEST 542               SPECIAL DEST, MODE 7, TEST
                                ;*******************************************************************************
107600                          TS542:

107600  012701  107722                  MOV     #YYBTP1,R1              ;SET UP THE DATA BUFFER.
107604  012700  107732                  MOV     #YYBTP2,R0
107610  012702  000004                  MOV     #4,R2
107614  012021                  1$:     MOV     (R0)+,(R1)+
107616  077202                          SOB     R2,1$
107620  012700  102541                  MOV     #YYBTP3-5201,R0
107624  012760  107722  005201          MOV     #YYBTP1,5201(R0)
107632  042737  100000  107722          BIC     #100000,@#YYBTP1                ;MAKE THE OPERAND POSITIVE.
107640  012701  000200                  MOV     #200,R1                 ;SET FD.
107644  170101                          LDFPS   R1

107646  005001                          CLR     R1
107650  170770  005201          YYB2:   NEGD    @5201(R0)                       ;TEST INSTRUCTION.

107654  170205                          STFPS   R5                      ;GET FPS.
107656  005701                          TST     R1                      ;WAS THE PC CORRECT AFTER EXECUTION?
107660  001031                          BNE     YYB10
107662  012701  107722                  MOV     #YYBTP1,R1                      ;IS THE RESULT CORRECT.
107666  012702  107732                  MOV     #YYBTP2,R2
107672  012703  000004                  MOV     #4,R3
107676  022122                  1$:     CMP     (R1)+,(R2)+
107700  001021                          BNE     YYB10                   ;BRANCH IF INCORRECT.
107702  077303                          SOB     R3,1$
107704  022700  102541                  CMP     #YYBTP3-5201,R0 ;IS R0 CORRECT.
107710  001015                          BNE     YYB10                   ;BRANCH IF INCORRECT.
107712  022705  000210                  CMP     #210,R5                 ;IS THE FPS CORRECT?
107716  001012                          BNE     YYB10                   ;BRANCH IF INCORRECT.
107720  000412                          BR      YYBDONE

                                ;THESE ARE DATA TABLES AND A DATA BUFFER.
107722  023245                  YYBTP1: 023245
107724  026720                          26720
107726  122324                          122324
107730  052672                          52672
107732  123245                  YYBTP2: 123245
107734  026720                          26720
107736  123324                          123324
107740  052672                          52672
107742  107722                  YYBTP3: YYBTP1
```

```
107744                          YYB10:
107744  104000                          EMT                             ;

107746                          YYBDONE:
107746  004767  010522                  JSR     PC,.RSET                ;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?).
                                ;*****************************************************************
                                ;TEST 543       NEGD, ABSD AND TSTD TEST
                                ;*****************************************************************
107752                          TS543:
                                ;TEST NEGD WITH POS NONZERO OPERAND
107752  004767  000526          WWB1:   JSR     PC,NATSUB
107756  000000                  1$:     0                               ;FLAG=NEGD.
107760  016341                  2$:     16341                           ;OPERAND.
107762  055772                          55772
107764  021133                          21133
107766  055447                          55447
107770  116341                  3$:     116341                          ;RESULT.
107772  055772                          55772
107774  021133                          21133
107776  055447                          55447
110000  016341                  4$:     16341                           ;ERROR RES.
110002  055772                          55772
110004  021133                          21133
110006  055447                          55447
110010  000207                  5$:     207                             ;FPS BEFORE EXECUTION.
110012  000210                          210                             ;FPS AFTER EXECUTION.
110014  000200                          200                             ;ERROR FPS.
110016  177777                          -1                              ;FEC
                                ;TEST NEGD WITH NEG OPERAND.
110020  004767  000460          WWB2:   JSR     PC,NATSUB
110024  000000                  1$:     0                               ;FLAG=NEGD.
110026  152525                  2$:     152525                          ;OPERAND.
110030  053545                          53545
110032  055565                          55565
110034  057505                          57505
110036  052525                  3$:     52525                           ;RESULT.
110040  053545                          53545
110042  055565                          55565
110044  057505                          57505
110046  152525                  4$:     152525                          ;ERROR RES.
110050  053545                          53545
110052  055565                          55565
110054  057505                          57505
110056  000217                  5$:     217                             ;FPS BEFORE EXECUTION.
110060  000200                          200                             ;FPS AFTER EXECUTION.
110062  000210                          210                             ;ERROR FPS.
110064  177777                          -1                              ;FEC
                                ;TEST ABSD WITH POSITIVE OPERAND
110066  004767  000412          WWB3:   JSR     PC,NATSUB
110072  000001                  1$:     1                               ;FLAG=ABSD.
110074  060705                  2$:     60705                           ;OPERAND.
110076  124735                          124735
```

```
110100 060124                   60124
110102 073560                   73560
110104 060705           3$:     60705                   ;RESULT.
110106 124735                   124735
110110 060124                   60124
110112 073560                   73560
110114 160705           4$:     160705                  ;ERROR RES.
110116 124735                   124735
110120 060124                   60124
110122 073560                   73560
110124 000217           5$:     217                     ;FPS BEFORE EXECUTION.
110126 000200                   200                     ;FPS AFTER EXECUTION.
110130 000210                   210                     ;ERROR FPS.
110132 177777                   -1                      ;EITHER BUT OP1B

                        ;TEST ABSD WITH NEG. OPERAND
110134 004767 000344    WWB4:   JSR     PC,NATSUB
110140 000001           1$:     1                       ;FLAG=ABSD.
110142 154345           2$:     154345                  ;OPERAND.
110144 076567                   76567
110146 032123                   32123
110150 043234                   43234
110152 054345           3$:     54345                   ;RESULT.
110154 076567                   76567
110156 032123                   32123
110160 043234                   43234
110162 154345           4$:     154345                  ;ERROR RES.
110164 076567                   76567
110166 032123                   32123
110170 043234                   43234
110172 000217           5$:     217                     ;FPS BEFORE EXECUTION.
110174 000200                   200                     ;FPS AFTER EXECUTION.
110176 177777                   -1                      ;ERROR FPS.
110200 177777                   -1

                        ;TEST WITH POSITIVE OP
110202 004767 000276    WWB5:   JSR     PC,NATSUB
110206 000002           1$:     2                       ;FLAG=TSTD.
110210 012321           2$:     12321                   ;OPERAND.
110212 045654                   45654
110214 070107                   70107
110216 034543                   34543
110220 012321           3$:     12321                   ;RESULT.
110222 045654                   45654
110224 070107                   70107
110226 034543                   34543
110230 112321           4$:     112321                  ;ERROR RES.
110232 045654                   45654
110234 070107                   70107
110236 034543                   34543
110240 000217           5$:     217                     ;FPS BEFORE EXECUTION.
110242 000200                   200                     ;FPS AFTER EXECUTION.
110244 000210                   210                     ;ERROR FPS.
110246 177777                   -1

                        ;TEST TSTD WITH NEG OP
110250 004767 000230    WWB6:   JSR     PC,NATSUB
110254 000002           1$:     2                       ;FLAG=TSTD.
110256 123765           2$:     123765                  ;OPERAND.
```

```
110260 023407                 23407
110262 034510                 34510
110264 045621                 45621
110266 123765          3$:    123765                  ;RESULT.
110270 023407                 23407
110272 034510                 34510
110274 045621                 45621
110276 023765          4$:    23765                   ;ERROR RES.
110300 023407                 23407
110302 034510                 34510
110304 045621                 45621
110306 000207          5$:    207                     ;FPS BEFORE EXECUTION.
110310 000210                 210                     ;FPS AFTER EXECUTION.
110312 000200                 200                     ;ERROR FPS.
110314 177777                 -1
                       ;TEST TSTD 0 OP
110316 004767 000162   WWB7:  JSR     PC,NATSUB
110322 000002          1$:    2                       ;FLAG=TSTD.
110324 000175          2$:    175                     ;OPERAND.
110326 176737                 176737
110330 071727                 71727
110332 037574                 37574
110334 000175          3$:    175                     ;RESULT.
110336 176737                 176737
110340 071727                 71727
110342 037574                 37574
110344 000000          4$:    0                       ;ERROR RES.
110346 000000                 0
110350 000000                 0
110352 000000                 0
110354 000200          5$:    200                     ;FPS BEFORE EXECUTION.
110356 000204                 204                     ;FPS AFTER EXECUTION.
110360 000214                 214                     ;ERROR FPS.
110362 177777                 -1
                       ;TEST TSTD -0 OP FIUV=0
110364 004767 000114   WWB10: JSR     PC,NATSUB
110370 000002          1$:    2                       ;FLAG=TSTD.
110372 100123          2$:    100123                  ;OPERAND.
110374 021012                 21012
110376 034565                 34565
110400 043210                 43210
110402 100123          3$:    100123                  ;RESULT.
110404 021012                 21012
110406 034565                 34565
110410 043210                 43210
110412 000000          4$:    0                       ;ERROR RES.
110414 000000                 0
110416 000000                 0
110420 000000                 0
110422 040203          5$:    40203                   ;FPS BEFORE EXECUTION.
110424 040214                 040214                  ;FPS AFTER EXECUTION.
110426 140214                 140214                  ;ERROR FPS.
110430 177777                 -1
                       ;TEST TSTD -0 OP FIUV=1
110432 004767 000046   WWB11: JSR     PC,NATSUB
110436 000002          1$:    2                       ;FLAG=TSTD.
```

```
110440  100137                  2$:     100137                  ;OPERAND.
110442  024613                          24613
110444  057024                          57024
110446  060137                          60137
110450  100137                  3$:     100137                  ;RESULT.
110452  024613                          24613
110454  057024                          57024
110456  060137                          60137
110460  000000                  4$:     0                       ;ERROR RES.
110462  000000                          0
110464  000000                          0
110466  000000                          0
110470  044200                  5$:     44200                   ;FPS BEFORE EXECUTION.
110472  144214                          144214                  ;FPS AFTER EXECUTION.
110474  044214                          044214                  ;ERROR FPS.
110476  000014                          14
110500  000167  000162                  JMP     WWBDONE

                                ;THIS SUBROUTINE, NATSUB, IS USED TO SET UP THE OPERANDS, EXECUTE
                                ;THE EITHER A TSTD, AN ABSD OR A NEGD INSTRUCTION AND CHECK THE RESULTS. A CALL
                                ;TO IT IS MADE THUS:
                                ;
                                ;               JSR     PC,@#NATSUB
                                ;               FLAG:   .WORD   X               ;INSTRUCTION TYPE FLAG.
                                ;               ACARG:  .WORD   X,X,X,X         ;OPERAND
                                ;               RES:    .WORD   X,X,X,X         ;EXPECTED RESULT
                                ;               ERRES:  .WORD   X,X,X,X         ;ERROR RESULT
                                ;               FPSB:   .WORD   X               ;FPS BEFORE EXECUTION
                                ;               FPSA:   .WORD   X               ;FPS AFTER EXECUTION
                                ;               FEC:    .WORD   X               ;EXPECTED FEC
                                ;               ERFPS:  .WORD   X               ;ERROR FPS.
                                ;               ERR1:   ERROR   X               ;DATA ERROR.
                                ;                       BR      CONT
                                ;               ERR2:   ERROR   X               ;FPS ERROR.
                                ;               CONT:                           ;RETURN ADDRESS
                                ;
                                ;THE OPERAND IS SET UP IN NATBF1. THEN
                                ;THE EITHER THE TSTD, NEGD OR ABSD INSTRUCTION IS EXECUTED.
                                ;NATSUB USES THE FIRST OPERAND AS A FLAG TO DETERMINE WHICH INSTRUCTION
                                ;IS TO BE EXECUTED: 0 = NEGD, 1 = ABSD, 2 = TSTD.
                                ;THE RESULT IS CHECKED AGAINST RES. IF THE RESULT IS CORRECT THEN THE FPS IS
                                ;COMPARED WITH FPSA. IF THIS TOO IS CORRECT NATSUB RETURNS CONTROL
                                ;TO THE CALLING ROUTINE AT CONT. IF THE FPS IS BAD NATSUB
                                ;COMPARE IT TO ERROR FPS. IF THIS MATCHES THEN NATSUB WILL RETURN
                                ;TO THE ERROR CALL AT ERR2. OTHERWISE NATSUB ITSELF
                                ;REPORTS THIS FAILURE AND THEN RETURNS TO CONT. IF THE RESULT OF THE
                                ;INSTRUCTION IS INCORRECT. THE INCORRECT RESULT IS COMPARED WITH THE
                                ;ANTICIPATED FAILING DATA PATTERN, ERRES. IF THE FAILURE IN
                                ;THE RESULT WAS ANTICIPATED CORRECTLY TO BE ERRES THEN NATSUB
                                ;WILL TRANSFER CONTROL TO THE ERROR CALL AT ERR1. OTHERWISE THE
                                ;RESULT WAS INCORRECT BUT WAS NOT ANTICIPATED AND NATSUB WILL
                                ;REPORT THE FAILURE AFTER WHICH CONTROL WILL BE PASSED TO CONT.


110504  012601                  NATSUB: MOV     (SP)+,R1                ;GET A POINTER TO THE ARGUMENTS.
110506  010102                          MOV     R1,R2           ;COPY THE OPERAND.
```

```
110510 062702 000002                  ADD     #2,R2
110514 012703 110654                  MOV     #NATBF1,R3
110520 012704 000004                  MOV     #4,R4
110524 012223                 1$:     MOV     (R2)+,(R3)+
110526 077402                         SOB     R4,1$
110530 016100 000032                  MOV     32(R1),R0           ;LOAD THE FPS.
110534 170100                         LDFPS   R0
110536 012700 110654                  MOV     #NATBF1,R0          ;SET UP THE OPERAND ADDRESS.
110542 011102                         MOV     (R1),R2             ;GET THE FLAG TO DETERMINE WHICH
110544 006302                         ASL     R2                  ;INSTRUCTION TO EXECUTE.
110546 006302                         ASL     R2                  ;0 = NEGD, 1 = ABSD, 2 = TSTD
110550 012703 110560                  MOV     #NATINS,R3
110554 060203                         ADD     R2,R3
110556 000113                         JMP     (R3)                ;GO EXECUTE THE INSTRUCTION.
110560 170710                 NATINS: NEGD    (R0)
110562 000403                         BR      2$
110564 170610                         ABSD    (R0)
110566 000401                         BR      2$
110570 170510                         TSTD    (R0)

110572 170204                 2$:     STFPS   R4                  ;GET THE FPS.
110574 170305                         STST    R5                  ;GET THE FEC.
110576 010100                         MOV     R1,R0               ;WAS THE RESULT CORRECT?
110600 062700 000012                  ADD     #12,R0
110604 012702 110654                  MOV     #NATBF1,R2
110610 012703 000004                  MOV     #4,R3
110614 022022                 3$:     CMP     (R0)+,(R2)+
110616 001014                         BNE     10$                 ;BRANCH IF INCORRECT.
110620 077303                         SOB     R3,3$
110622 026104 000034                  CMP     34(R1),R4           ;WAS THE FPS CORRECT?
110626 001010                         BNE     10$                 ;BRANCH IF INCORRECT.
110630 005761 000034                  TST     34(R1)              ;IF THE EXPECTED FPS WAS NEGATIVE CHECK THE FEC.
110634 100003                         BPL     4$
110636 026105 000040                  CMP     40(R1),R5           ;WAS THE FEC CORRECT.
110642 001002                         BNE     10$                 ;BRANCH IF INCORRECT.
110644 000161 000042          4$:     JMP     42(R1)              ;RETURN.

110650                        10$:
110650 104000                         EMT                         ;

110652 177777                         .WORD   -1
110654 177777 177777 177777   NATBF1: .WORD   -1,-1,-1,-1,-1
110662 177777 177777

110666                        WWBDONE:
110666 004767 007602                  JSR     PC,.RSET            ;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?).




                              ;*******************************************************************
                              ;TEST 544        SOURCE MODES, MODE 1 (FL=0), TEST
                              ;*******************************************************************
```

```
110672                             TS544:



110672  012700  110742                     MOV     #AACTP1,R0      ;SET UP TEST DATA IN BUFFER.
110676  012710  147517                     MOV     #147517,(R0)
110702  012737  110716  037366             MOV     #AAC2,@#$TMP2
110710  012737  110746  000004             MOV     #AAC10,@#ERRVECT        ;SET UP FOR TRAPS TO 4.
110716  170110                     AAC2:   LDFPS   (R0)            ;TEST INSTRUCTION.

110720  170205                             STFPS   R5              ;GET FPS

110722  020027  110742                     CMP     R0,#AACTP1      ;IS R0 CORRECT?
110726  001007                             BNE     AAC10           ;BR IF NOT.
110730  022705  147517                     CMP     #147517,R5      ;IS FPS CORRECT?
110734  001004                             BNE     AAC10           ;BR IF NOT.
110736  000404                             BR      AACDONE

                                   ;TEST BUFFER AND DATA:
110740  177777                             -1
110742  147517                     AACTP1: 147517
110744  177777                             -1
110746                             AAC10:
110746  104000                             EMT                     ;

110750                             AACDONE:
110750  004767  007520                     JSR     PC,.RSET        ;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?).


                                   ;***********************************************************************
                                   ;TEST 545        SOURCE MODES, MODE 2 (FL=0), TEST
                                   ;***********************************************************************
110754                             TS545:


110754  012700  111016                     MOV     #BBCTP1,R0      ;SET UP TEST DATA IN BUFFER.
110760  012710  145212                     MOV     #145212,(R0)
110764  012737  111022  000004             MOV     #BBC10,@#ERRVECT ;SET UP FOR TRAPS TO 4.

110772  170120                     BBC2:   LDFPS   (R0)+           ;TEST INSTRUCTION.

110774  170205                             STFPS   R5              ;GET FPS

110776  020027  111020                     CMP     R0,#BBCTP1+2    ;IS R0 CORRECT?
111002  001007                             BNE     BBC10           ;BR IF NOT.
111004  022705  145212                     CMP     #145212,R5      ;IS THE FPS CORRECT?
111010  001004                             BNE     BBC10           ;BR IF NOT.
111012  000404                             BR      BBCDONE


                                   ;TEST BUFFER AND DATA:
111014  177777                             -1
```

```
111016  177777                    BBCTP1: .WORD   -1
111020  177777                                    -1


111022                            BBC10:
111022  104000                            EMT                             ;
111024                            BBCDONE:
111024  004767  007444                    JSR     PC,.RSET                ;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?).


                                  ;*****************************************************************************
                                  ;TEST 546       SOURCE MODES, MODE 4 (FL=0), TEST
                                  ;*****************************************************************************
111030                            TS546:


111030  012700  111112                    MOV     #DDCTP1+2,R0            ;SET UP THE TEST DATA BUFFER.
111034  012760  105252  177776            MOV     #105252,-2(R0)
111042  012737  111056  037366            MOV     #DDC2,@#$TMP2
111050  012737  111122  000004            MOV     #DDC10,@#ERRVEC
111056  170140                    DDC2:   LDFPS   -(R0)
111060  170205                            STFPS   R5
111062  020027  111110                    CMP     R0,#DDCTP1
111066  001015                            BNE     DDC10
111070  022705  105252                    CMP     #105252,R5
111074  001012                            BNE     DDC10
111076  000412                            BR      DDCDONE

111100  177777  177777  177777            -1,-1,-1,-1
111106  177777
111110  177777                    DDCTP1: -1
111112  177777  177777  177777            -1,-1,-1,-1
111120  177777
111122                            DDC10:
111122  104000                            EMT                             ;
111124                            DDCDONE:
111124  004767  007344                    JSR     PC,.RSET                ;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?).
                                  ;*****************************************************************************
                                  ;TEST 547       SOURCE MODES, MODE 3 (FL=0), TEST
                                  ;*****************************************************************************
111130                            TS547:
111130  012700  111216                    MOV     #EECTP2,R0
111134  012710  111206                    MOV     #EECTP1,(R0)
111140  012767  103456  000040            MOV     #103456,EECTP1
111146  012737  111230  000004            MOV     #EEC10,@#ERRVECT ;SET UP FOR TRAPS TO 4.
111154  170130                    EEC2:   LDFPS   @(R0)+                  ;TEST INSTRUCTION.
111156  170205                            STFPS   R5                      ;GET THE FPS.
111160  020027  111220                    CMP     R0,#EECTP2+2            ;IS R0 CORRECT?
```

```
111164  001021                          BNE     EEC10           ;BR IF NOT.
111166  022705  103456                  CMP     #103456,R5      ;IS THE FPS CORRECT?
111172  001016                          BNE     EEC10           ;BR IF NOT.
111174  000416                          BR      EECDONE


                                ;TEST BUFFER AND DATA:
111176  177777  177777  177777          -1,-1,-1,-1
111204  177777
111206  177777                  EECTP1: -1
111210  177777  177777  177777          -1,-1,-1
111216  111206  177777  177777  EECTP2: EECTP1,-1,-1,-1,
111224  177777  000000

111230                          EEC10:
111230  104000                          EMT                     ;
111232                          EECDONE:
111232  004767  007236                  JSR     PC,.RSET        ;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?).
                                ;*************************************************************************
                                ;TEST 550       SOURCE MODES, MODE 5 (FL=0), TEST
                                ;*************************************************************************
111236                          TS550:
111236  012700  111322                  MOV     #FFCTP2+2,R0    ;SET UP THE TEST DATA BUFFER.
111242  012760  111310  177776          MOV     #FFCTP1,-2(R0)
111250  012737  045412  111310          MOV     #45412,@#FFCTP1
111256  012737  111330  000004          MOV     #FFC10,@#ERRVECT ;SET UP FOR TRAPS TO 4.
111264  170150                  FFC2:   LDFPS   @-(R0)          ;TEST INSTRUCTION.
111266  170205                          STFPS   R5              ;GET THE FPS.
111270  020027  111320                  CMP     R0,#FFCTP2      ;IS R0 CORRECT?
111274  001015                          BNE     FFC10           ;BR IF NOT.
111276  022705  045412                  CMP     #45412,R5       ;IS THE FPS CORRECT?
111302  001012                          BNE     FFC10           ;BR IF NOT.
111304  000412                          BR      FFCDONE


                                ;TEST BUFFER AND DATA:
111306  177777                          -1
111310  177777                  FFCTP1: -1
111312  177777  177777  177777          -1,-1,-1
111320  111310  177777  177777  FFCTP2: FFCTP1,-1,-1,-1
111326  177777

111330                          FFC10:
111330  104000                          EMT                     ;
111332                          FFCDONE:
111332  004767  007136                  JSR     PC,.RSET        ;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?).
                                ;*************************************************************************
                                ;TEST 551       SOURCE MODES, MODE 6 (FL=0), TEST
```

```
                                      ;*****************************************************************************
111336                                TS551:
111336  012700  104211                        MOV     #GGCTP1-5201,R0 ;SET UP THE TEST DATA BUFFER.
111342  012737  046543  111412                MOV     #46543,@#GGCTP1
111350  005001                                CLR     R1
111352  012737  111424  000004                MOV     #GGC10,@#ERRVECT ;SET UP FOR TRAPS TO 4.
111360  170160  005201                GGC2:   LDFPS   5201(R0)                  ;TEST INSTRUCTION.
111364  170204                                STFPS   R4              ;GET THE FPS.
111366  005701                                TST     R1              ;WAS PC CORRECT AFTER EXECUTION?
111370  001015                                BNE     GGC10           ;BR IF NOT.
111372  020027  104211                        CMP     R0,#GGCTP1-5201 ;IS R0 CORRECT?
111376  001012                                BNE     GGC10           ;BR IF NOT.
111400  022704  046543                        CMP     #46543,R4       ;IS THE FPS CORRECT?
111404  001007                                BNE     GGC10           ;BR IF NOT.
111406  000407                                BR      GGCDONE


                                      ;TEST BUFFER AND DATA:
111410  177777                                -1
111412  177777  177777  177777        GGCTP1: -1,-1,-1,-1
111420  177777
111422  177777                                -1
111424                                GGC10:
111424  104000                                EMT                     ;
111426                                GGCDONE:
111426  004767  007042                        JSR     PC,.RSET        ;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?).
                                      ;*****************************************************************************
                                      ;TEST 552       SOURCE MODES, MODE 7 (FL=0), TEST
                                      ;*****************************************************************************
111432                                TS552:
111432  012700  104323                        MOV     #HHCTP2-5201,R0 ;SET UP THE TEST DATA BUFFER.
111436  012760  111514  005201                MOV     #HHCTP1,5201(R0)
111444  012737  004547  111514                MOV     #4547,@#HHCTP1
111452  005001                                CLR     R1
111454  012737  111534  000004                MOV     #HHC10,@#ERRVECT ;SET UP FOR TRAPS TO 4.
111462  170170  005201                HHC2:   LDFPS   @5201(R0)                 ;TEST INSTRUCTION.
111466  170204                                STFPS   R4              ;GET THE FPS.
111470  005701                                TST     R1              ;WAS PC CORRECT AFTER EXECUTION?
111472  001020                                BNE     HHC10           ;BR IF NOT.
111474  020027  104323                        CMP     R0,#HHCTP2-5201 ;IS R0 CORRECT?
111500  001015                                BNE     HHC10           ;BR IF NOT.
111502  022704  004547                        CMP     #4547,R4        ;IS THE FPS CORRECT?
111506  001012                                BNE     HHC10           ;BR IF NOT.
111510  000412                                BR      HHCDONE


                                      ;TEST BUFFER AND DATA:
111512  177777                                -1
111514  177777  177777  177777        HHCTP1: .WORD -1,-1,-1,-1
111522  177777
111524  177777  177777  177777        HHCTP2: .WORD -1,-1,-1,-1
111532  177777
```

```
111534                          HHC10:
111534  104000                          EMT                             ;
111536                          HHCDONE:
111536  004767  006732                  JSR     PC,.RSET                ;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?).



                                ;*************************************************************************
                                ;TEST 553       SOURCE MODES, MODE 2 GR7 (FL=1), TEST
                                ;*************************************************************************
111542                          TS553:

111542  012737  111600  000004          MOV     #IIC20,@#ERRVECT ;SET UP FOR TRAPS TO 4.
111550  012700  000300                  MOV     #300,R0
111554  170100                          LDFPS   R0
111556  005001                          CLR     R1

111560  177027                  IIC2:   LDCLD   (R7)+,AC0               ;TEST INSTRUCTION.
111562  005201                          5201
111564  005201                          5201
111566  005201                          5201
111570  005201                          5201

111572  020127  000003                  CMP     R1,#3                   ;WAS PC CORRECT AFTER EXECUTION?
111576  001401                          BEQ     IICDONE
111600                          IIC20:
111600  104000                          EMT                             ;

111602                          IICDONE:
111602  004767  006666                  JSR     PC,.RSET                ;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?).



                                ;*************************************************************************
                                ;TEST 554       SOURCE MODES, MODE 2 (FL=1), TEST
                                ;*************************************************************************
111606                          TS554:

111606  012700  000300                  MOV     #300,R0
111612  170100                          LDFPS   R0
111614  012700  111660                  MOV     #TCCBF0,R0              ;SET UP THE TEST DATA BUFFER.
111620  177020                  TCC2:   LDCLD   (R0)+,AC0               ;TEST INSTRUCTION.

111622  170204                          STFPS   R4                      ;GET THE FPS.
111624  012701  111670                  MOV     #TCCBF1,R1              ;GET THE RESULT.
111630  012702  000200                  MOV     #200,R2
111634  170102                          LDFPS   R2
111636  174011                          STD     AC0,(R1)
111640  020027  111664                  CMP     R0,#TCCBF0+4            ;IS R0 CORRECT?
```

```
111644  001401                      BEQ     TCC3
111646  104000                      EMT                             ;

111650  022704  000300      TCC3:   CMP     #300,R4 ;IS THE FPS CORRECT?
111654  001411                      BEQ     TCCDONE
111656  104000                      EMT                             ;


                            ;TEST BUFFER AND DATA:
111660  001234  067076  054321  TCCBF0: .WORD   01234,67076,54321,012345
111666  012345
111670  177777  177777  177777  TCCBF1: -1,-1,-1,-1
111676  177777

111700                      TCCDONE:
111700  004767  006570              JSR     PC,.RSET        ;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?).



                            ;*******************************************************************************
                            ;TEST 555       LDCIF AND LDCLF TEST
                            ;*******************************************************************************
111704                      TS555:


                            ;ZERO   OPERAND FL=0

111704  004737  112576      KKC1:   JSR     PC,@#LDCFSUB    ;GO EXECUTE INSTRUCTION.

111710  000000  000000      1$:     .WORD   0,0             ;FSRC OPERAND.
111714  000000  000000      2$:     .WORD   0,0             ;EXPECTED RESULT.
111720  177777  177777      3$:     .WORD   -1,-1           ;ANTICIPATED ERRONEOUS RESULT.
111724  000000              4$:     0                       ;FPS BEFORE EXECUTION.
111726  000004                      4                       ;FPS AFTER EXECUTION.
111730  177777                      -1                      ;ANTICIPATED ERRONEOUS FPS.
                            ;ZERO   OPERAND FL=0

111732  004737  112576      KKC2:   JSR     PC,@#LDCFSUB    ;GO EXECUTE THE INSTRUCTION.

111736  000000  177777      1$:     .WORD   0,-1            ;FSRC OPERAND.
111742  000000  000000      2$:     .WORD   0,0             ;EXPECTED RESULT.
111746  004177  177400      3$:     4177,177400             ;ANTICIPATED ERRONEOUS RESULT.
111752  000000              4$:     0                       ;FPS BEFORE EXECUTION.
111754  000004                      4                       ;FPS AFTER EXECUTION.
111756  177777                      -1                      ;ANTICIPATED ERRONEOUS FPS.
                            ;ZERO   OPERAND FL=1

111760  004737  112576      KKC3:   JSR     PC,@#LDCFSUB    ;GO EXECUTE THE INSTRUCTION.

111764  000000  000000      1$:     .WORD   0,0             ;FSRC OPERAND.
111770  000000  000000      2$:     .WORD   0,0             ;EXPECTED RESULT.
111774  177777  177777      3$:     .WORD   -1,-1           ;ANTICIPATED ERRONEOUS RESULT.
```

```
20318 112000 000100                 4$:     100                     ;FPS BEFORE EXECUTION.
20319 112002 000104                         104                     ;FPS AFTER EXECUTION.
20320 112004 000004                         4                       ;ANTICIPATED ERRONEOUS FPS.
20321                               ;OPERAND        POSITIVE        FL=0
20322 112006 004737 112576          KKC4:   JSR     PC,@#LDCFSUB    ;GO EXECUTE THE INSTRUCTION.
20323 112012 040000 000000          1$:     .WORD   40000,0         ;FSRC OPERAND.
20324 112016 043600 000000          2$:     .WORD   43600,0         ;EXPECTED RESULT.
20325 112022 047600 000000          3$:     .WORD   47600,0         ;ANTICIPATED ERRONEOUS RESULT.
20326 112026 000017                 4$:     17                      ;FPS BEFORE EXECUTION.
20327 112030 000000                         0                       ;FPS AFTER EXECUTION.
20328 112032 177777                         -1                      ;ANTICIPATED ERRONEOUS FPS.
20329                               ;OPERAND=1,     FL=0
20330 112034 004737 112576          KKC5:   JSR     PC,@#LDCFSUB    ;GO EXECUTE THE INSTRUCTION.
20331 112040 000001 000000          1$:     .WORD   1,0             ;FSRC OPERAND.
20332 112044 040200 000000          2$:     .WORD   40200,0         ;EXPECTED RESULT.
20333 112050 044200 000000          3$:     .WORD   44200,0         ;ANTICIPATED ERRONEOUS RESULT.
20334 112054 000017                 4$:     17                      ;FPS BEFORE EXECUTION.
20335 112056 000000                         0                       ;FPS AFTER EXECUTION
20336 112060 177777                         -1                      ;ANTICIPATED ERRONEOUS FPS.
20337                               ;OPERAND=       PATTERN FL=0
20338 112062 004737 112576          KKC6:   JSR     PC,@#LDCFSUB    ;GO EXECUTE THE INSTRUCTION.
20339 112066 000252 000000          1$:     .WORD   252,0           ;FSRC OPERAND.
20340 112072 042052 000000          2$:     .WORD   42052,0         ;EXPECTED RESULT.
20341 112076 046052 000000          3$:     .WORD   46052,0         ;ANTICIPATED ERRONEOUS RESULT.
20342 112102 000000                 4$:     0                       ;FPS BEFORE EXECUTION.
20343 112104 000000                         0                       ;FPS AFTER EXECUTION.
20344 112106 177777                         -1                      ;ANTICIPATED ERRONEOUS FPS.
20345                               ;OPERAND=-40000 FL=0
20346 112110 004737 112576          KKC7:   JSR     PC,@#LDCFSUB    ;GO EXECUTE THE INSTRUCTION.
20347 112114 140000 000000          1$:     .WORD   -40000,0                ;FSRC OPERAND.
20348 112120 143600 000000          2$:     .WORD   143600,0                ;EXPECTED RESULT.
20349 112124 043600 000000          3$:     .WORD   43600,0         ;ANTICIPATED ERRONEOUS RESULT.
20350 112130 000007                 4$:     7                       ;FPS BEFORE EXECUTION.
20351 112132 000010                         10                      ;FPS AFTER EXECUTION.
20352 112134 177777                         -1                      ;ANTICIPATED ERRONEOUS FPS.
20353                               ;OPERAND=-1     FL-0
20354 112136 004737 112576          KKC10:  JSR     PC,@#LDCFSUB    ;GO EXECUTE THE INSTRUCTION.
20355 112142 177777 000000          1$:     .WORD   -1,0            ;FSRC OPERAND.
20356 112146 140200 000000          2$:     .WORD   140200,0                ;EXPECTED RESULT.
20357 112152 144000 000400          3$:     .WORD   144000,400              ;ANTICIPATED ERRONEOUS RESULT.
20358 112156 000000                 4$:     0                       ;FPS BEFORE EXECUTION.
20359 112160 000010                         10                      ;FPS AFTER EXECUTION.
20360 112162 177777                         -1                      ;ANTICIPATED ERRONEOUS FPS.
20361                               ;OPERAND=PATTERN        FL=0
20362 112164 004737 112576          KKC11:  JSR     PC,@#LDCFSUB    ;GO EXECUTE THE INSTRUCTION.
20363 112170 125252 000000          1$:     .WORD   125252,0                ;FSRC OPERAND.
20364 112174 143652 126000          2$:     .WORD   143652,126000           ;EXPECTED RESULT.
20365 112200 043652 126000          3$:     .WORD   43652,126000            ;ANTICIPATED ERRONEOUS RESULT.
20366 112204 000007                 4$:     7                       ;FPS BEFORE EXECUTION.
20367 112206 000010                         10                      ;FPS AFTER EXECUTION.
20368 112210 177777                         -1                      ;ANTICIPATED ERRONEOUS FPS.
20369                               ;OPERAND        POS     FL-1
20370 112212 004737 112576          KKC12:  JSR     PC,@#LDCFSUB    ;GO EXECUTE THE INSTRUCTION.
20371 112216 040000 000000          1$:     .WORD   40000,0         ;FSRC OPERAND.
20372 112222 047600 000000          2$:     .WORD   47600,0         ;EXPECTED RESULT.
20373 112226 043600 000000          3$:     .WORD   43600,0         ;ANTICIPATED ERRONEOUS RESULT.
```

```
112232 000117                4$:     117                ;FPS BEFORE EXECUTION.
112234 000100                        100                ;FPS AFTER EXECUTION.
112236 177777                        -1                 ;ANTICIPATED ERRONEOUS FPS.
                             ;OPERAND=1         FL=1
112240 004737 112576         KKC13:  JSR     PC,@#LDCFSUB       ;GO EXECUTE THE INSTRUCTION.
112244 000000 000001         1$:     .WORD   0,1                ;FSRC OPERAND.
112250 040200 000000         2$:     .WORD   40200,0            ;EXPECTED RESULT.
112254 034200 000000         3$:     .WORD   34200,0            ,ANTICIPATED ERRONEOUS RESULT.
112260 000100                4$:     100                ;FPS BEFORE EXECUTION.
112262 000100                        100                ;FPS AFTER EXECUTION.
112264 177777                        -1                 ;ANTICIPATED ERRONEOUS FPS.
                             ;OPERAND=          PATTERN FL=1
112266 004737 112576         KKC14:  JSR     PC,@#LDCFSUB       ;GO EXECUTE THE INSTRUCTION.
112272 000000 000252         1$:     .WORD   0,252              ;FSRC OPERAND.
112276 042052 000000         2$:     .WORD   42052,0            ;EXPECTED RESULT.
112302 036052 000000         3$:     .WORD   36052,0            ;ANTICIPATED ERRONEOUS RESULT.
112306 000111                4$:     111                ;FPS BEFORE EXECUTION.
112310 000100                        100                ;FPS AFTER EXECUTION.
112312 177777                        -1                 ;ANTICIPATED ERRONEOUS FPS.
                             ;OPERAND=-40000,0          FL=1
112314 004737 112576         KKC15:  JSR     PC,@#LDCFSUB       ;GO EXECUTE THE INSTRUCTION.
112320 140000 000000         1$:     .WORD   -40000,0                   ;FSRC OPERAND.
112324 147600 000000         2$:     .WORD   147600,0                   ;EXPECTED RESULT.
112330 047600 000000         3$:     .WORD   47600,0            ;ANTICIPATED ERRONEOUS RESULT.
112334 000107                4$:     107                ;FPS BEFORE EXECUTION.
112336 000110                        110                ;FPS AFTER EXECUTION.
112340 177777                        -1                 ;ANTICIPATED ERRONEOUS FPS.
                             ;OPERAND=-1,-1     FL=1
112342 004737 112576         KKC16:  JSR     PC,@#LDCFSUB       ;GO EXECUTE THE INSTRUCTION.
112346 177777 177777         1$:     .WORD   -1,-1              ;FSRC OPERAND.
112352 140200 000000         2$:     .WORD   140200,0                   ;EXPECTED RESULT.
112356 150000 000000         3$:     .WORD   150000,0                   ;ANTICIPATED ERRONEOUS RESULT.
112362 000100                4$:     100                ;FPS BEFORE EXECUTION.
112364 000110                        110                ;FPS AFTER EXECUTION.
112366 177777                        -1                 ;ANTICIPATED ERRONEOUS FPS.
                             ;OPERAND=-PATTERN          FL=1,   ROUND MODE
112370 004737 112576         KKC17:  JSR     PC,@#LDCFSUB       ;GO EXECUTE THE INSTRUCTION.
112374 125252 125252         1$:     .WORD   125252,125252              ;FSRC OPERAND.
112400 147652 125253         2$:     .WORD   147652,125253              ;EXPECTED RESULT.
112404 047652 125253         3$:     .WORD   47652,125253               ;ANTICIPATED ERRONEOUS RESULT.
112410 000105                4$:     105                ;FPS BEFORE EXECUTION.
112412 000110                        110                ;FPS AFTER EXECUTION.
112414 177777                        -1                 ;ANTICIPATED ERRONEOUS FPS.
                             ;OPERAND=77777,177500      FL=1,   ROUND MODE
112416 004737 112576         KKC20:  JSR     PC,@#LDCFSUB       ;GO EXECUTE THE INSTRUCTION.
112422 077777 177500         1$:     .WORD   77777,177500               ;FSRC OPERAND.
112426 047777 177777         2$:     .WORD   47777,177777               ;EXPECTED RESULT.
112432 047777 177776         3$:     .WORD   47777,177776               ;ANTICIPATED ERRONEOUS RESULT.
112436 000117                4$:     117                ;FPS BEFORE EXECUTION.
112440 000100                        100                ;FPS AFTER EXECUTION.
112442 177777                        -1                 ;ANTICIPATED ERRONEOUS FPS.
                             ;OPERAND=40000,000100      FL=1,   ROUND MODE
112444 004737 112576         KKC21:  JSR     PC,@#LDCFSUB       ;GO EXECUTE THE INSTRUCTION.
112450 040000 000100         1$:     .WORD   40000,100                  ;FSRC OPERAND.
112454 047600 000001         2$:     .WORD   47600,1            ;EXPECTED RESULT.
112460 047600 000000         3$:     .WORD   47600,0            ;ANTICIPATED ERRONEOUS RESULT.
```

```
112464  000102                  4$:     102                             ;FPS BEFORE EXECUTION.
112466  000100                          100                             ;FPS AFTER EXECUTION.
112470  177777                          -1                              ;ANTICIPATED ERRONEOUS FPS.
                                ;OPERAND=40000,000100   FL=1,   TRUNC MODE
112472  004737  112576          KKC22:  JSR     PC,@#LDCFSUB    ;GO EXECUTE THE INSTRUCTION.
112476  040000  000100          1$:     .WORD   40000,100               ;FSRC OPERAND.
112502  047600  000000          2$:     .WORD   47600,0         ;EXPECTED RESULT.
112506  047600  000001          3$:     .WORD   47600,1         ;ANTICIPATED ERRONEOUS RESULT.
112512  000157                  4$:     157                             ;FPS BEFORE EXECUTION.
112514  000140                          140                             ;FPS AFTER EXECUTION.
112516  177777                          -1                              ;ANTICIPATED ERRONEOUS FPS.
                                ;OPERAND=100000,0 (MOST NEG #)  FL=0
112520  004737  112576          KKC23:  JSR     PC,@#LDCFSUB    ;GO EXECUTE THE INSTRUCTION.
112524  100000  000000          1$:     .WORD   100000,0                ;FSRC OPERAND.
112530  144000  000000          2$:     .WORD   144000,0                ;EXPECTED RESULT.
112534  143600  000000          3$:     .WORD   143600,0                ;ANTICIPATED ERRONEOUS RESULT.
112540  000007                  4$:     7                               ;FPS BEFORE EXECUTION.
112542  000010                          10                              ;FPS AFTER EXECUTION
112544  177777                          -1                              ;ANTICIPATED ERRONEOUS FPS.
                                ;OPERAND=100000,0       FL=1
112546  004737  112576          KKC24:  JSR     PC,@#LDCFSUB    ;GO EXECUTE THE INSTRUCTION.
112552  100000  000000          1$:     .WORD   100000,0                ;FSRC OPERAND.
112556  150000  000000          2$:     .WORD   150000,0                ;EXPECTED RESULT.
112562  147600  000000          3$:     .WORD   147600,0                ;ANTICIPATED ERRONEOUS RESULT.
112566  000107                  4$:     107                             ;FPS BEFORE EXECUTION.
112570  000110                          110                             ;FPS AFTER EXECUTION.
112572  177777                          -1                              ;ANTICIPATED ERRONEOUS FPS.
112574  000441                  6$:     BR      KKCDONE

                                ;THIS SUBROUTINE, LDCFSUB, IS USED TO SET UP THE OPERANDS, EXECUTE
                                ;THE LDCIF OR LDCLF INSTRUCTION AND CHECK THE RESULTS. A CALL
                                ;TO IT IS MADE THUS:
                                ;
                                ;               JSR     PC,@#LDCFSUB
                                ;               ACARG:  .WORD   X,X             ;AC OPERAND
                                ;               RES:    .WORD   X,X             ;EXPECTED RESULT
                                ;               ERRES:  .WORD   X,X             ;ERROR RESULT
                                ;               FPSB:   .WORD   X               ;FPS BEFORE EXECUTION
                                ;               FPSA:   .WORD   X               ;FPS AFTER EXECUTION
                                ;               ERFPS:  .WORD   X               ;ERROR FPS
                                ;               ERR1:   ERROR   X               ;DATA ERROR
                                ;                       BR      CONT
                                ;               ERR2:   ERROR   X               ;FPS ERROR
                                ;               CONT:                           ;RETURN ADDRESS
                                ;
                                ;THE OPERANDS ARE SET UP (USING AC0 AS THE ACCUMULATOR). THEN
                                ;THE LDCIF OR LDCLF INSTRUCTION IS EXECUTED.
                                ;THE RESULT IS CHECKED AGAINST RES. IF THE RESULT IS CORRECT THEN THE FPS IS
                                ;COMPARED WITH FPSA IF THIS TOO IS CORRECT LDCFSUB RETURNS CONTROL
                                ;TO THE CALLING ROUTINE AT CONT. IF THE FPS IS BAD LDCFSUB WILL
                                ;COMPARE IT TO ERROR FPS. IF THIS MATCHES THEN LDCFSUB WILL RETURN
                                ;TO THE ERROR CALL AT ERR2, OTHERWISE LDCFSUB ITSELF
                                ;REPORTS THIS FAILURE AND THEN RETURNS TO CONT. IF THE RESULT OF THE
                                ;LDCIF OR LDCLF IS INCORRECT, THE INCORRECT RESULT IS COMPARED WITH THE
                                ;ANTICIPATED FAILING DATA PATTERN, ERRES. IF THE FAILURE IN
                                ;THE RESULT WAS ANTICIPATED CORRECTLY TO BE ERRES THEN LDCFSUB
```

```
                                        ;WILL TRANSFER CONTROL TO THE ERROR CALL AT ERR1. OTHERWISE THE
                                        ;RESULT WAS INCORRECT BUT WAS NOT ANTICIPATED AND LDCFSUB
                                        ;REPORT THE FAILURE AFTER WHICH CONTROL WILL BE PASSED TO CONT.

112576  012601                          LDCFSUB:        MOV     (SP)+,R1        ;GET A POINTER TO THE ARGUMENTS.
112600  016100  000014                          MOV     14(R1),R0       ;SET THE FPS.
112604  170100                                  LDFPS   R0
112606  010100                                  MOV     R1,R0

112610  177010                          1$:     LDCIF   (R0),AC0                ;TEST INSTRUCTION LDCIF OR LDCLF.

112612  170204                                  STFPS   R4              ;GET FPS.
112614  012700  112670                          MOV     #LDCT,R0        ;GET THE RESULT.
112620  012702  000200                          MOV     #200,R2
112624  170102                                  LDFPS   R2
112626  174010                                  STD     AC0,(R0)

112630  012702  112670                          MOV     #LDCT,R2        ;SEE IF THE RESULT WAS CORRECT.
112634  010100                                  MOV     R1,R0
112636  062700  000004                          ADD     #4,R0
112642  012703  000002                          MOV     #2,R3
112646  022022                          2$:     CMP     (R0)+,(R2)+
112650  001006                                  BNE     10$             ;BR IF INCORRECT.
112652  077303                                  SOB     R3,2$

112654  026104  000016                          CMP     16(R1),R4       ;SEE IF THE FPS WAS CORRECT.
112660  001002                                  BNE     10$             ;BR IF INCORRECT.
112662  000161  000022                  3$:     JMP     22(R1)          ;RETURN.
112666                                  10$:
112666  104000                                  EMT                     ;

                                        ;DATA BUFFER:
112670  000000  000000  000000          LDCT:   .WORD   0,0,0,0
112676  000000

112700                                  KKCDONE:
112700  004767  005570                          JSR     PC,.RSET        ;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?).



                                        ;***************************************************************************
                                        ;TEST 556       LDCID AND LDCLD TEST
                                        ;***************************************************************************
112704                                  TS556:
                                        ;OPERAND=0      FL=0,   FD=1
112704  004737  113362                  LLC1:   JSR     PC,@#LDCDSUB    ;GO EXECUTE THE INSTRUCTION.
112710  000000  000000                  1$:     .WORD   0,0             ;FSRC OPERAND.
112714  000000  000000  000000          2$:     .WORD   0,0,0,0         ;EXPECTED RESULT.
112722  000000
112724  177777  177777  177777          3$:     .WORD   -1,-1,-1,-1             ;ANTICIPATED ERRONEOUS RESULT.
112732  177777
112734  000213                          4$:     213                     ;FPS BEFORE EXECUTION.
112736  000204                                  204                     ;FPS AFTER EXECUTION.
```

```
112740 177777                                -1                           ;ANTICIPATED ERRONEOUS FPS.
                                     ;OPERAND=0     FL=0,   FD=1
112742 004737 113362                 LLC2:   JSR    PC,@#LDCDSUB          ;GO EXECUTE THE INSTRUCTION.
112746 000000 177777                 1$:     .WORD  0,-1                  ;FSRC OPERAND.
112752 000000 000000 000000          2$:     .WORD  0,0,0,0               ;EXPECTED RESULT.
112760 000000
112762 004177 177400 000000          3$:     .WORD  4177,177400,0,0            ;ANTICIPATED ERRONEOUS RESULT.
112770 000000
112772 000200                        4$:     200                          ;FPS BEFORE EXECUTION.
112774 000204                                204                          ;FPS AFTER EXECUTION.
112776 177777                                -1                           ;ANTICIPATED ERRONEOUS FPS.
                                     ;OPERAND=0     FL=1    FD=1
113000 004737 113362                 LLC3:   JSR    PC,@#LDCDSUB          ;GO EXECUTE THE INSTRUCTION.
113004 000000 000000                 1$:     .WORD  0,0                   ;FSRC OPERAND.
113010 000000 000000 000000          2$:     .WORD  0,0,0,0               ;EXPECTED RESULT.
113016 000000
113020 177777 177777 177777          3$:     .WORD  -1,-1,-1,-1                ;ANTICIPATED ERRONEOUS RESULT.
113026 177777
113030 000211                        4$:     211                          ;FPS BEFORE EXECUTION.
113032 000204                                204                          ;FPS AFTER EXECUTION.
113034 177777                                -1                           ;ANTICIPATED ERRONEOUS FPS.
                                     ;OPERAND=40000 FL=0    FD=1
113036 004737 113362                 LLC4:   JSR    PC,@#LDCDSUB          ;GO EXECUTE THE INSTRUCTION.
113042 040000 000000                 1$:     .WORD  40000,0               ;FSRC OPERAND.
113046 043600 000000 000000          2$:     .WORD  43600,0,0,0                ;EXPECTED RESULT.
113054 000000
113056 047600 000000 000000          3$:     .WORD  47600,0,0,0                ;ANTICIPATED ERRONEOUS RESULT.
113064 000000
113066 000217                        4$:     217                          ;FPS BEFORE EXECUTION.
113070 000200                                200                          ;FPS AFTER EXECUTION.
113072 177777                                -1                           ;ANTICIPATED ERRONEOUS FPS.
                                     ;OPERAND=-40000 FL=0   FD=1
113074 004737 113362                 LLC5:   JSR    PC,@#LDCDSUB          ;GO EXECUTE THE INSTRUCTION.
113100 140000 000000                 1$:     .WORD  -40000,0                   ;FSRC OPERAND.
113104 143600 000000 000000          2$:     .WORD  143600,0,0,0               ;EXPECTED RESULT.
113112 000000
113114 043600 000000 000000          3$:     .WORD  43600,0,0,0                ;ANTICIPATED ERRONEOUS RESULT.
113122 000000
113124 000200                        4$:     200                          ;FPS BEFORE EXECUTION.
113126 000210                                210                          ;FPS AFTER EXECUTION.
113130 177777                                -1                           ;ANTICIPATED ERRONEOUS FPS.
                                     ;OPERAND=40000,0         FL=1        FD=1
113132 004737 113362                 LLC6:   JSR    PC,@#LDCDSUB          ;GO EXECUTE THE INSTRUCTION.
113136 040000 000000                 1$:     .WORD  40000,0               ;FSRC OPERAND.
113142 047600 000000 000000          2$:     .WORD  47600,0,0,0                ;EXPECTED RESULT.
113150 000000
113152 043600 000000 000000          3$:     .WORD  43600,0,0,0                ;ANTICIPATED ERRONEOUS RESULT.
113160 000000
113162 000317                                317                          ;FPS BEFORE EXECUTION.
113164 000300                                300                          ;FPS AFTER EXECUTION.
113166 177777                                -1                           ;ANTICIPATED ERRONEOUS FPS.
                                     ;OPERAND=0,1   FL=1    FD=1
113170 004737 113362                 LLC7:   JSR    PC,@#LDCDSUB          ;GO EXECUTE THE INSTRUCTION.
113174 000000 000001                 1$:     .WORD  0,1                   ;FSRC OPERAND.
113200 040200 000000 000000          2$:     .WORD  40200,0,0,0                ;EXPECTED RESULT.
113206 000000
```

```
113210  034200  000000  000000  3$:     .WORD   34200,0,0,0             ;ANTICIPATED ERRONEOUS RESULT.
113216  000000
113220  000300                  4$:     300                     ;FPS BEFORE EXECUTION.
113222  000300                          300                     ;FPS AFTER EXECUTION.
113224  177777                          -1                      ;ANTICIPATED ERRONEOUS FPS.
                                ;OPERAND=77777,177777   FL=1    FD=1
113226  004737  113362          LLC10:  JSR     PC,@#LDCDSUB    ;GO EXECUTE THE INSTRUCTION.
113232  077777  177777          1$:     .WORD   77777,177777            ;FSRC OPERAND.
113236  047777  177777  177000  2$:     .WORD   47777,177777,177000,0           ;EXPECTED RESULT.
113244  000000
113246  177777  177777  177777  3$:     .WORD   -1,-1,-1,-1             ;ANTICIPATED ERRONEOUS RESULT.
113254  177777
113256  000317                  4$:     317                     ;FPS BEFORE EXECUTION.
113260  000300                          300                     ;FPS AFTER EXECUTION.
113262  177777                          -1                      ;ANTICIPATED ERRONEOUS FPS.
                                ;OPERAND=-PATTERN       FL=1    FD=1

113264  004767  000072          LLC11:  JSR     PC,LDCDSUB      ;GO EXECUTE THE INSTPUCTION.
113270  177777  177526          1$:     .WORD   -1,-252         ;FSRC OPERAND.
113274  142052  000000  000000  2$:     .WORD   142052,0,0,0            ;EXPECTED RESULT.
113302  000000
113304  136052  000000  000000  3$:     .WORD   136052,0,0,0            .ANTICIPATED ERRONEOUS RESULT.
113312  000000
113314  000307                  4$:     307                     ;FPS BEFORE EXECUTION.
113316  000310                          310                     ;FPS AFTER EXECUTION.
113320  177777                          -1                      ;ANTICIPATED ERRONEOUS FPS.
                                ;OPERAND=PATTERN        FL=1    FD=1    FT=1
113322  004767  000034          LLC12:  JSR     PC,LDCDSUB      ;GO EXECUTE THE INSTRUCTION.
113326  012345  067012          1$:     .WORD   12345,67012             ;FSRC OPERAND.
113332  047247  025560  050000  2$:     .WORD   47247,025560,050000,0           ;EXPECTED RESULT.
113340  000000
113342  177777  177777  177777  3$:     .WORD   -1,-1,-1,-1             ;ANTICIPATED ERRONEOUS RESULT.
113350  177777
113352  000352                  4$:     352                     ;FPS BEFORE EXECUTION.
113354  000340                          340                     ;FPS AFTER EXECUTION.
113356  177777                          -1                      ;ANTICIPATED ERRONEOUS FPS.
113360  000435                  6$:     BR      LLCDONE

                                ;THIS SUBROUTINE, LDCDSUB, IS USED TO SET UP THE OPERANDS, EXECUTE
                                ;THE LDCID OR LDCLD INSTRUCTION AND CHECK THE RESULTS. A CALL
                                ;TO IT IS MADE THUS:
                                ;
                                ;               JSR     PC,@#LDCDSUB
                                ;               ACARG:  .WORD   X,X             ;AC OPERAND
                                ;               RES:    .WORD   X,X,X,X         ;EXPECTED RESULT
                                ;               ERRES:  .WORD   X,X,X,X         ;ERROR RESULT
                                ;               FPSB:   .WORD   X               ;FPS BEFORE EXECUTION
                                ;               FPSA:   .WORD   X               ;FPS AFTER EXECUTION
                                ;               ERFPS:  .WORD   X               ;ERROR FPS.
                                ;               ERR1:   ERROR   X               ;DATA ERROR.
                                ;                       BR      CONT
                                ;               ERR2:   ERROR   X               ;FPS ERROR.
                                ;               CONT:                           ;RETURN ADDRESS
                                ;
                                ;THE OPERANDS ARE SET UP (USING AC0 AS THE ACCUMULATOR). THEN
                                ;THE LDCID OR LDCLD INSTRUCTION IS EXECUTED.
```

```
                                 ;THE RESULT IS CHECKED AGAINST RES. IF THE RESULT IS CORRECT THEN THE FPS IS
                                 ;COMPARED WITH FPSA IF THIS TOO IS CORRECT LDCDSUB RETURNS CONTROL
                                 ;TO THE CALLING ROUTINE AT CONT. IF THE FPS IS BAD LDCDSUB
                                 ;COMPARE IT TO ERROR FPS. IF THIS MATCHES THEN LDCDSUB WILL RETURN
                                 ;TO THE ERROR CALL AT ERR2, OTHERWISE LDCDSUB ITSELF
                                 ;REPORTS THIS FAILURE AND THEN RETURNS TO CONT. IF THE RESULT OF THE
                                 ;LDCID OR LDCLD IS INCORRECT, THE INCORRECT RESULT IS COMPARED WITH THE
                                 ;ANTICIPATED FAILING DATA PATTERN, ERRES. IF THE FAILURE IN
                                 ;THE RESULT WAS ANTICIPATED CORRECTLY TO BE ERRES THEN LDCDSUB
                                 ;WILL TRANSFER CONTROL TO THE ERROR CALL AT ERR1. OTHERWISE THE
                                 ;RESULT WAS INCORRECT BUT WAS NOT ANTICIPATED AND LDCDSUB WILL
                                 ;REPORT THE FAILURE AFTER WHICH CONTROL WILL BE PASSED TO CONT.

113362  012601                   LDCDSUB:        MOV     (SP)+,R1        ;GET A POINTER TO THE ARGUMENTS.
113364  016100  000024                   MOV     24(R1),R0       ;SET THE FPS.
113370  170100                           LDFPS   R0
113372  010100                           MOV     R1,R0
113374  177010                   1$:     LDCID   (R0),AC0        ;TEST INSTRUCTION, LDCID OR LDCLD.

113376  170204                           STFPS   R4              ;GET FPS.
113400  012700  112670                   MOV     #LDCT,R0        ;GET THE RESULT.
113404  012702  000200                   MOV     #200,R2
113410  170102                           LDFPS   R2
113412  174010                           STD     AC0,(R0)

                                 ;SEE IF THE RESULT IS CORRECT.
113414  012702  112670                   MOV ,   #LDCT,R2
113420  010100                           MOV     R1,R0
113422  062700  000004                   ADD     #4,R0
113426  012703  000002                   MOV     #2,R3
113432  022022                   2$:     CMP     (R0)+,(R2)+
113434  001006                           BNE     10$             ;BR IF INCORRECT.
113436  077303                           SOB     R3,2$

113440  026104  000026                   CMP     26(R1),R4       ;IS THE FPS CORRECT?
113444  001002                           BNE     10$             ;BR IF INCORRECT.
113446  000161  000032           3$:     JMP     32(R1)          ;RETURN.
113452                           10$:
113452  104000                           EMT                     ;

113454                           LLCDONE:
113454  004767  005014                   JSR     PC,.RSET        ;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?).


                                 ;*************************************************************************
                                 ;TEST 557        LDEXP TEST
                                 ;*************************************************************************
113460                           TS557:

                                 ; NON-ZERO RES. VALID EXPON=210 (EXCESS 200)=10
113460  004767  001136           MMC1:   JSR     PC,LDXSUB       ;GO EXECUTE THE INSTRUCTION.
113464  012345  067012  034567   1$:     .WORD   12345,67012,34567,012345        ;AC0 OPERAND.
```

```
113472  012345
113474  000010                  2$:     .WORD   10              ;EXPONENT OPERAND.
113476  042145  067012  034567  3$:     .WORD   42145,67012,34567,012345                ;EXPECTED RESULT.
113504  012345
113506  002145  067012  034567  4$:     .WORD   2145,67012,34567,012345         ;ANTICIPATED ERRONEOUS RESULT.
113514  012345
113516  047217                  5$:     47217                   ;FPS BEFORE EXECUTION.
113520  047200                          47200                   ;FPS AFTER EXECUTION.
113522  147200                          147200                  ;ANTICIPATED ERRONEOUS FPS.
113524  177777                          -1              ;EXPECTED FEC.
                                ;NON-ZERO       RES     NEG.
113526  004737  114622          MMC2:   JSR     PC,@#LDXSUB     ;EXPON=377
113532  123456  070123  045670  1$:     .WORD   123456,70123,45670,123456       ;ACO OPERAND.
113540  123456
113542  000177                  2$:     .WORD   177             ;EXPONENT OPERAND.
113544  177656  070123  045670  3$:     .WORD   177656,70123,45670,123456               ;EXPECTED RESULT.
113552  123456
113554  137656  070123  045670  4$:     .WORD   137656,70123,45670,123456               ;ANTICIPATED ERRONEOUS RESULT.
113562  123456
113564  047207                  5$:     47207                   ;FPS BEFORE EXECUTION.
113566  047210                          47210                   ;FPS AFTER EXECUTION.
113570  147210                          147210                  ;ANTICIPATED ERRONEOUS FPS.
113572  177777                          -1              ;EXPECTED FEC.
                                ;NON-ZERO       RES,    EXP=256=(56)REAL
113574  004737  114622          MMC3:   JSR     PC,@#LDXSUB     ;GO EXECUTE THE INSTRUCTION.
113600  073261  057645  043323  1$:     .WORD   73261,057645,43323,101760       ;ACO OPERAND.
113606  101760
113610  000056                  2$:     .WORD   56              ;EXPONENT OPERAND.
113612  053461  057645  043323  3$:     .WORD   53461,057645,43323,101760               ;EXPECTED RESULT.
113620  101760
113622  177777  177777  177777  4$:     .WORD   -1,-1,-1,-1             ;ANTICIPATED ERRONEOUS RESULT.
113630  177777
113632  047200                  5$:     47200                   ;FPS BEFORE EXECUTION.
113634  047200                          47200                   ;FPS AFTER EXECUTION.
113636  147200                          147200                  ;ANTICIPATED ERRONEOUS FPS.
113640  177777                          -1              ;EXPECTED FEC.
                                ;EXP=27 (EXCESS 200)=-151 (OCT)
113642  004737  114622          MMC4:   JSR     PC,@#LDXSUB     ;GO EXECUTE THE INSTRUCTION.
113646  012223  024252  062720  1$:     .WORD   12223,24252,62720,21222 ;ACO OPERAND.
113654  021222
113656  177627                  2$:     .WORD   -151            ;EXPONENT OPERAND.
113660  005623  024252  062720  3$:     .WORD   5623,24252,62720,21222          ;EXPECTED RESULT.
113666  021222
113670  177777  177777  177777  4$:     .WORD   -1,-1,-1,-1             ;ANTICIPATED ERRONEOUS RESULT.
113676  177777
113700  047200                  5$:     47200                   ;FPS BEFORE EXECUTION.
113702  047200                          47200                   ;FPS AFTER EXECUTION.
113704  147200                          147200                  ;ANTICIPATED ERRONEOUS FPS.
113706  177777                          -1              ;EXPECTED FEC.
                                ;EXP=0 (EXCESS 200)=-200 (OCT), POSITIVE FRAC
                                ; FIV=1
113710  004737  114622          MMC5:   JSR     PC,@#LDXSUB     ;GO EXECUTE THE INSTRUCTION.
113714  030131  032334  035363  1$:     .WORD   30131,32334,35363,73031 ;ACO OPERAND.
113722  073031
113724  177600                  2$:     .WORD   -200            ;EXPONENT OPERAND.
113726  000131  032334  035363  3$:     .WORD   00131,32334,35363,73031         ;EXPECTED RESULT.
```

```
20766  113734  073031
20767  113736  000000  000000  000000  4$:     .WORD   0,0,0,0                 ;ANTICIPATED ERRONEOUS RESULT.
20768  113744  000000
20769  113746  042200                  5$:     42200                           ;FPS BEFORE EXECUTION.
20770  113750  142204                          142204                          ;FPS AFTER EXECUTION.
20771  113752  042202                          42202                           ;ANTICIPATED ERRONEOUS FPS.
20772  113754  000012                          12                      ;EXPECTED FEC.
20773                                  ;EXP=0  (EXCESS 200)=-200 (OCT), NEG FRACT,FIU=1
20774  113756  004737  114622          MMC6:   JSR     PC,@#LDXSUB     ;GO EXECUTE THE INSTRUCTION.
20775  113762  140414  024344  045464  1$:     .WORD   140414,24344,45464,74045        ;AC0 OPERAND.
20776  113770  074045
20777  113772  177600                  2$:     .WORD   -200                    ;EXPONENT OPERAND.
20778  113774  100014  024344  045464  3$:     .WORD   100014,24344,45464,74045        ;-0             ;EXPECTED RESULT.
20779  114002  074045
20780  114004  000000  000000  000000  4$:     .WORD   0,0,0,0                 ;ANTICIPATED ERRONEOUS RESULT.
20781  114012  000000
20782  114014  042200                  5$:     42200                           ;FPS BEFORE EXECUTION.
20783  114016  142214                          142214                          ;FPS AFTER EXECUTION
20784  114020  042214                          42214                           ;ANTICIPATED ERRONEOUS FPS.
20785  114022  000012                          12                      ;EXPECTED FEC.
20786                                  ;EXP=0 (EXCESS 200)=-200 (OCT),POS FRAC, FIU=0
20787
20788  114024  004737  114622          MMC7:   JSR     PC,@#LDXSUB     ;GO EXECUTE THE INSTRUCTION.
20789  114030  051525  035455  005675  1$:     .WORD   51525,35455,5675,05152  ;AC0 OPERAND.
20790  114036  005152
20791  114040  177600                  2$:     .WORD   -200                    ;EXPONENT OPERAND.
20792  114042  000000  000000  000000  3$:     .WORD   0,0,0,0                 ;EXPECTED RESULT.
20793  114050  000000
20794  114052  000125  035455  005675  4$:     .WORD   00125,35455,5675,05152          ;ANTICIPATED ERRONEOUS RESULT.
20795  114060  005152
20796  114062  045200                          45200                           ;FPS BEFORE EXECUTION.
20797  114064  045204                          45204                           ;FPS AFTER EXECUTION.
20798  114066  145204                          145204                          ;ANTICIPATED ERRONEOUS FPS.
20799  114070  177777                          -1                      ;EXPECTED FEC.
20800                                  ;EXP=-1405 (EXCESS 200)=-1605 (OCT), FIU=1
20801  114072  004737  114622          MMC10:  JSR     PC,@#LDXSUB     ;GO EXECUTE THE INSTRUCTION.
20802  114076  061626  062636  046566  1$:     .WORD   61626,62636,46566,67606 ;AC0 OPERAND.
20803  114104  067606
20804  114106  176173                  2$:     .WORD   -1605                   ;EXPONENT OPERAND.
20805  114110  076626  062636  046566  3$:     .WORD   76626,62636,46566,67606         ;EXPECTED RESULT.
20806  114116  067606
20807  114120  000000  000000  000000  4$:     .WORD   0,0,0,0                 ;ANTICIPATED ERRONEOUS RESULT.
20808  114126  000000
20809  114130  042200                  5$:     42200                           ;FPS BEFORE EXECUTION.
20810  114132  142200                          142200                          ;FPS AFTER EXECUTION.
20811  114134  042204                          42204                           ;ANTICIPATED ERRONEOUS FPS.
20812  114136  000012                          12                      ;EXPECTED FEC.
20813                                  ;EXP=-17416 (EXCESS 200)=-17616 (OCT), FIU=0
20814  114140  004737  114622          MMC11:  JSR     PC,@#LDXSUB     ;GO EXECUTE THE INSTRUCTION.
20815  114144  071727  037475  076777  1$:     .WORD   71727,37475,76777,17273 ;AC0 OPERAND.
20816  114152  017273
20817  114154  160162                  2$:     .WORD   -17616                  ;EXPONENT OPERAND.
20818  114156  000000  000000  000000  3$:     .WORD   0,0,0,0                 ;EXPECTED RESULT.
20819  114164  000000
20820  114166  074527  037475  076777  4$:     .WORD   74527,37475,76777,17273         ;ANTICIPATED ERRONEOUS RESULT.
20821  114174  017273
```

```
114176  045200                  5$:     45200                   ;FPS BEFORE EXECUTION.
114200  045204                          45204                   ;FPS AFTER EXECUTION.
114202  145200                          145200                  ;ANTICIPATED ERRONEOUS FPS.
114204  177777                          -1              ;EXPECTED FEC.
                                ;EXP=-1601 (EXCESS 200)=-2001 (OCT), FIU=1
114206  004737  114622          MMC12:  JSR     PC,@#LDXSUB     ;GO EXECUTE THE INSTRUCTION.
114212  001020  030405  006070  1$:     .WORD   01020,30405,06070,00102 ;ACO OPERAND.
114220  000102
114222  175777                  2$:     .WORD   -2001           XPONENT OPERAND.
114224  037620  030405  006070  3$:     .WORD   37620,30405,06  )0102           ;EXPECTED RESULT.
114232  000102
114234  000000  000000  000000  4$:     .WORD   0,0,0,0         ,NTICIPATED ERRONEOUS RESULT.
114242  000000
114244  042200                  5$:     42200                   ;FPS BEFORE EXECUTION.
114246  142200                          142200                  ;FPS AFTER EXECUTION.
114250  042204                          42204                   ;ANTICIPATED ERRONEOUS FPS.
114252  000012                          12              ;EXPECTED FEC.
                                ;EXP=1206 (EXCESS 200)=1006 (OCT) FIV =1
114254  004737  114622          MMC13:  JSR     PC,@#LDXSUB     ;GO EXECUTE THE INSTRUCTION.
114260  012131  014151  016171  1$:     .WORD   12131,14151,16171,10111 ;ACO OPERAND.
114266  010111
114270  001006                  2$:     .WORD   1006            ;EXPONENT OPERAND.
114272  041531  014151  016171  3$:     .WORD   41531,14151,16171,10111         ;EXPECTED RESULT.
114300  010111
114302  000000  000000  000000  4$:     .WORD   0,0,0,0         ;ANTICIPATED ERRONEOUS RESULT.
114310  000000
114312  041200                  5$:     41200                   ;FPS BEFORE EXECUTION.
114314  141202                          141202                  ;FPS AFTER EXECUTION.
114316  041204                          41204                   ;ANTICIPATED ERRONEOUS FPS.
114320  000010                          10              ;EXPECTED FEC.
                                ;EXP=16315 (EXCESS 200)=16115 (OCT) FIV=0
114322  004737  114622          MMC14:  JSR     PC,@#LDXSUB     ;GO EXECUTE THE INSTRUCTION.
114326  027262  025242  023222  1$:     .WORD   27262,25242,23222,21202 ;ACO OPERAND.
114334  021202
114336  016115                  2$:     .WORD   16115           ;EXPONENT OPERAND.
114340  000000  000000  000000  3$:     .WORD   0,0,0,0         ;EXPECTED RESULT.
114346  000000
114350  063262  025242  023222  4$:     .WORD   63262,25242,23222,21202         ;ANTICIPATED ERRONEOUS RESULT.
114356  021202
114360  046200                  5$:     46200                   ;FPS BEFORE EXECUTION.
114362  046206                          46206                   ;FPS AFTER EXECUTION.
114364  146202                          146202                  ;ANTICIPATED ERRONEOUS FPS.
114366  177777                          -1              ;EXPECTED FEC.
                                ;EXP=11011 (EXCESS 200)=10611 (OCT) FIV=1

114370  004737  114622          MMC15:  JSR     PC,@#LDXSUB     ;GO EXECUTE THE INSTRUCTION.
114374  030313  032333  034353  1$:     .WORD   30313,32333,34353,36373 ;ACO OPERAND.
114402  036373
114404  010611                  2$:     .WORD   10611           ;EXPONENT OPERAND.
114406  002313  032333  034353  3$:     .WORD   2313,32333,34353,36373          ;EXPECTED RESULT.
114414  036373
114416  000000  000000  000000  4$:     .WORD   0,0,0,0         ;ANTICIPATED ERRONEOUS RESULT.
114424  000000
114426  041200                  5$:     41200                   ;FPS BEFORE EXECUTION.
114430  141202                          141202                  ;FPS AFTER EXECUTION.
114432  041204                          41204                   ;ANTICIPATED ERRONEOUS FPS.
```

```
114434  000010                                  10                    ;EXPECTED FEC.
                                        ;EXP=17123 (EXCESS 200)=16723 (OCT) FIV=0

114436  004737  114622                  MMC16:  JSR     PC,@#LDXSUB     ;GO EXECUTE THE INSTRUCTION.
114442  040414  042434  044454          1$:     .WORD   40414,42434,44454,46474 ;ACO OPERAND.
114450  046474
114452  016723                          2$:     .WORD   16723           ;EXPONENT OPERAND.
114454  000000  000000  000000          3$:     .WORD   0,0,0,0         ;EXPECTED RESULT.
114462  000000
114464  024614  042434  044454          4$:     .WORD   24614,42434,44454,46474         ;ANTICIPATED ERRONEOUS RESULT.
114472  046474
114474  046200                          5$.     46200                   ;FPS BEFORE EXECUTION.
114476  046206                                  46206                   ;FPS AFTER EXECUTION.
114500  146202                                  146202                  ;ANTICIPATED ERRONEOUS FPS.
114502  177777                                  -1              ;EXPECTED FEC.
                                        ;EXP=   254 (OCT)=      454 (EXCESS 200)        FIV=1

114504  004737  114622                  MMC17:  JSR     PC,@#LDXSUB     ;GO EXECUTE THE INSTRUCTION.
114510  050515  052535  054555          1$:     .WORD   50515,52535,54555,56575 ;ACO OPERAND.
114516  056575
114520  000254                          2$:     .WORD   254             ;EXPONENT OPERAND.
114522  013115  052535  054555          3$:     .WORD   13115,52535,54555,56575         ;EXPECTED RESULT.
114530  056575
114532  000000  000000  000000          4$:     .WORD   0,0,0,0         ;ANTICIPATED ERRONEOUS RESULT.
114540  000000
114542  041200                          5$:     41200                   ;FPS BEFORE EXECUTION.
114544  141202                                  141202                  ;FPS AFTER EXECUTION.
114546  041204                                  41204                   ;ANTICIPATED ERRONEOUS FPS.
114550  000010                                  10              ;EXPECTED FEC.
                                        ;EXP=   313 (OCT)=      513(EXCESS 200) FIV=0

114552  004737  114622                  MMC20:  JSR     PC,@#LDXSUB     ;GO EXECUTE THE INSTRUCTION.
114556  060616  062636  064656          1$:     .WORD   60616,62636,64656,66676 ;ACO OPERAND.
114564  066676
114566  000313                          2$:     .WORD   313             ;EXPONENT OPERAND.
114570  000000  000000  000000          3$:     .WORD   0,0,0,0         ;EXPECTED RESULT.
114576  000000
114600  022616  062636  064656          4$:     .WORD   22616,62636,64656,66676         ;ANTICIPATED ERRONEOUS RESULT.
114606  066676
114610  046200                          5$:     46200                   ;FPS BEFORE EXECUTION.
114612  046206                                  46206                   ;FPS AFTER EXECUTION.
114614  146202                                  146202                  ;ANTICIPATED ERRONEOUS FPS.
114616  177777                                  -1              ;EXPECTED FEC.
114620  000457                                  BR      MMCDONE

                                        ;THIS SUBROUTINE, LDXSUB, IS USED TO SET UP THE OPERANDS, EXECUTE
                                        ;THE LDEXP INSTRUCTION AND CHECK THE RESULTS. A CALL
                                        ;TO IT IS MADE THUS:
                                        ;
                                        ;                       JSR     PC,@#LDXSUB
                                        ;                       ACARG:  .WORD   X,X,X,X         ;AC OPERAND
                                        ;                       EXP:    .WORD   X               ;EXPONENT
                                        ;                       RES:    .WORD   X,X,X,X         ;EXPECTED RESULT
                                        ;                       ERRES:  .WORD   X,X,X,X         ;ERROR RESULT
                                        ;                       FPSB:   .WORD   X               ;FPS BEFORE EXECUTION
                                        ;                       FPSA:   .WORD   X               ;FPS AFTER EXECUTION
```

```
                                ;               ERFPS:  .WORD   X               ;ERROR FPS.
                                ;               FEC:    .WORD   X               ;EXPECTED FEC
                                ;               ERR1:   ERROR   X               ;DATA ERROR.
                                ;                       BR      CONT
                                ;               ERR2:   ERROR   X               ;FPS ERROR.
                                ;               CONT:                           ;RETURN ADDRESS
                                ;
                                ;THE OPERANDS ARE SET UP (USING AC0 AS THE ACCUMULATOR). THEN
                                ;THE LDEXP INSTRUCTION IS EXECUTED.
                                ;THE RESULT IS CHECKED AGAINST RES. IF THE RESULT IS CORRECT THEN THE FPS IS
                                ;COMPARED WITH FPSA IF THIS TOO IS CORRECT LDXSUB RETURNS CONTROL
                                ;TO THE CALLING ROUTINE AT CONT. IF THE FPS IS BAD LDXSUB
                                ;COMPARE IT TO ERROR FPS. IF THIS MATCHES THEN LDXSUB WILL RETURN
                                ;TO THE ERROR CALL AT ERR2, OTHERWISE LDXSUB ITSELF
                                ;REPORTS THIS FAILURE AND THEN RETURNS TO CONT. IF THE RESULT OF THE
                                ;LDEXP IS INCORRECT, THE INCORRECT RESULT IS COMPARED WITH THE
                                ;ANTICIPATED FAILING DATA PATTERN, ERRES. IF THE FAILURE IN
                                ;THE RESULT WAS ANTICIPATED CORRECTLY TO BE ERRES THEN LDXSUB
                                ;WILL TRANSFER CONTROL TO THE ERROR CALL AT ERR1. OTHERWISE THE
                                ;RESULT WAS INCORRECT BUT WAS NOT ANTICIPATED AND LDXSUB WILL
                                ;REPORT THE FAILURE AFTER WHICH CONTROL WILL BE PASSED TO CONT.

114622  012601                  LDXSUB: MOV     (SP)+,R1        ;GET A POINTER TO THE ARGUMENTS.
114624  012700  000200                  MOV     #200,R0         ;LOAD THE AC0 OPERAND.
114630  170100                          LDFPS   R0
114632  010100                          MOV     R1,R0
114634  172410                          LDD     (R0),AC0
114636  016100  000032                  MOV     32(R1),R0       ;SET UP THE FPS.
114642  170100                          LDFPS   R0
114644  010100                          MOV     R1,R0
114646  062700  000010                  ADD     #10,R0

114652  176410                  1$:     LDEXP   (R0),AC0        ;TEST INSTRUCTION.

114654  170204                          STFPS   R4              ;GET THE FPS.
114656  170305                          STST    R5              ;GET THE FEC.
114660  012700  000200                  MOV     #200,R0         ;GET THE RESULT.
114664  170100                          LDFPS   R0
114666  012700  114750                  MOV     #LDXT,R0
114672  174010                          STD     AC0,(R0)
114674  012702  114750                  MOV     #LDXT,R2        ;SEE IF THE RESULT WAS CORRECT.
114700  010103                          MOV     R1,R3
114702  062703  000012                  ADD     #12,R3
114706  012700  000004                  MOV     #4,R0
114712  022223                  2$:     CMP     (R2)+,(R3)+
114714  001014                          BNE     10$             ;BRANCH IF NOT CORRECT.
114716  077003                          SOB     R0,2$
114720  020461  000034                  CMP     R4,34(R1)       ;SEE IF THE FPS WAS CORRECT.
114724  001010                          BNE     10$             ;BRANCH IF NOT CORRECT.
114726  005761  000034                  TST     34(R1)
114732  100003                          BPL     3$
114734  020561  000040                  CMP     R5,40(R1)       ;SEE IF THE FEC WAS CORRECT.
114740  001002                          BNE     10$             ;BRANCH IF NOT CORRECT.

114742  000161  000042          3$:     JMP     42(R1)          ;RETURN.
114746                          10$:
```

```
114746  104000                          EMT                             ;

                                ;DATA BUFFER:
114750  000000  000000  000000  LDXT:   .WORD   0,0,0,0
114756  000000

114760                          MMCDONE:
114760  004767  003510                  JSR     PC,.RSET        ;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?).



                                ;*************************************************************************
                                ;TEST 560       DESTINATION MODES, MODE 1 (FL=0), TEST
                                ;*************************************************************************
114764                          TS560:

114764  012700  115054                  MOV     #NNCTB0,R0      ;SET UP THF DATA BUFFER.
114770  012701  000006                  MOV     #6,R1
114774  012720  177777          1$:     MOV     #-1,(R0)+
115000  077103                          SOB     R1,1$
115002  012700  102345                  MOV     #102345,R0
115006  012737  115070  000004          MOV     #NNC10,@#ERRVECT ;SET UP FOR TRAPS TO 4.
115014  170100                          LDFPS   R0              ;SET UP FPS.
115016  012700  115060                  MOV     #NNCTB1,R0

115022  170210                  NNC2:   STFPS   (R0)            ;TEST INSTRUCTION.
115024  020027  115060                  CMP     R0,#NNCTB1      ;IS R0 CORRECT?
115030  001017                          BNE     NNC10           ;BRANCH IF NOT CORRECT.
115032  023727  115060  102345          CMP     @#NNCTB1,#102345  ;IS RESULT CORRECT?
115040  001013                          BNE     NNC10           ;BRANCH IF NOT CORRECT.
115042  023727  115062  177777          CMP     @#NNCTB1+2,#-1  ;IS THE RESULT CORRECT?
115050  001007                          BNE     NNC10           ;BRANCH IF NOT CORRECT.
115052  000407                          BR      NNCDONE

                                ;TEST DATA BUFFER:
115054  177777  177777          NNCTB0: .WORD   -1,-1
115060  177777  177777  177777  NNCTB1: .WORD   -1,-1,-1,-1
115066  177777
115070                          NNC10:
115070  104000                          EMT                     ;

115072                          NNCDONE:
115072  004767  003376                  JSR     PC,.RSET        ;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?).



                                ;*************************************************************************
                                ;TEST 561       DESTINATION MODES, MODE 2 (FL=0), TEST
```

```
                                        ;*************************************************************************
        115076                          TS561:

        115076  012700  115166                  MOV     #OOCTB0,R0      ;SET UP THE DATA BUFFER.
        115102  012701  000006                  MOV     #6,R1
        115106  012720  177777          1$:     MOV     #-1,(R0)+
        115112  077103                          SOB     R1,1$
        115114  012700  105412                  MOV     #105412,R0
        115120  012737  115202  000004          MOV     #OOC10,@#ERRVECT  ;SET UP FOR TRAPS TO VECTOR 4.
        115126  170100                          LDFPS   R0              ;SET UP FPS.
        115130  012700  115172                  MOV     #OOCTB1,R0

        115134  170220                  OOC2:   STFPS   (R0)+           ;TEST INSTRUCTION.
        115136  020027  115174                  CMP     R0,#OOCTB1+2    ;IS R0 CORRECT?
        115142  001017                          BNE     OOC10           ;BRANCH IF NOT CORRECT.
        115144  023727  115172  105412          CMP     @#OOCTB1,#105412  ;IS THE RESULT CORRECT?
        115152  001013                          BNE     OOC10           ;BRANCH IF NOT CORRECT.
        115154  023727  115174  177777          CMP     @#OOCTB1+2,#-1  ;IS THE RESULT CORRECT?
        115162  001007                          BNE     OOC10           ;BRANCH IF NOT CORRECT.
        115164  000407                          BR      OOCDONE

                                        ;TEST DATA BUFFER:
        115166  177777  177777          OOCTB0: .WORD   -1,-1
        115172  177777  177777  177777  OOCTB1: .WORD   -1,-1,-1,-1
        115200  177777
        115202                          OOC10:
        115202  104000                          EMT                     ;

        115204                          OOCDONE:
        115204  004767  003264                  JSR     PC,.RSET        ;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?).




                                        ;*************************************************************************
                                        ;TEST 562       DESTINATION MODES, MODE 4 (FL=0), TEST
                                        ;*************************************************************************
        115210                          TS562:

        115210  012700  115300                  MOV     #PPCTB0,R0      ;SET UP THE DATA BUFFER.
        115214  012701  000006                  MOV     #6,R1
        115220  012720  177777          1$:     MOV     #-1,(R0)+
        115224  077103                          SOB     R1,1$
        115226  012700  105555                  MOV     #105555,R0
        115232  012737  115314  000004          MOV     #PPC10,@#ERRVECT  ;SET UP FOR TRAPS TO VECTOR 4.
        115240  170100                          LDFPS   R0              ;SET UP FPS.
        115242  012700  115306                  MOV     #PPCTB1+2,R0

        115246  170240                  PPC2:   STFPS   -(R0)           ;TEST INSTRUCTION.
        115250  020027  115304                  CMP     R0,#PPCTB1      ;IS R0 CORRECT?
        115254  001017                          BNE     PPC10           ;BRANCH IF NOT CORRECT.
        115256  023727  115304  105555          CMP     @#PPCTB1,#105555  ;IS THE RESULT CORRECT?
```

```
115264  001013                                  BNE     PPC10             ;BRANCH IF NOT CORRECT.
115266  023727  115306  177777                  CMP     @#PPCTB1+2,#-1    ;IS THE RESULT CORRECT?
115274  001007                                  BNE     PPC10             ;BRANCH IF NOT CORRECT.
115276  000407                                  BR      PPCDONE

                                ;TEST DATA BUFFER:
115300  177777  177777          PPCTB0: .WORD   -1,-1
115304  177777  177777  177777  PPCTB1: .WORD   -1,-1,-1,-1
115312  177777
115314                          PPC10:
115314  104000                          EMT                               ;
115316                          PPCDONE:
115316  004767  003152                  JSR     PC,.RSET          ;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?).




                                ;*******************************************************************************
                                ;TEST 563       DESTINATION MODES, MODE 3 (FL=0), TEST
                                ;*******************************************************************************
115322                          TS563:

115322  012700  115416                  MOV     #QQCTB0,R0        ;SET UP THE DATA BUFFER.
115326  012701  000010                  MOV     #10,R1
115332  012720  177777          1$:     MOV     #-1,(R0)+
115336  077103                          SOB     R1,1$
115340  012700  106653                  MOV     #106653,R0
115344  012737  115436  000004          MOV     #QQC10,@#ERRVECT  ;SET UP FOR TRAPS TO VECTOR 4.
115352  170100                          LDFPS   R0                ;SET UP FPS.
115354  012700  115432                  MOV     #QQCTB2,R0
115360  012710  115422                  MOV     #QQCTB1,(R0)

115364  170230                  QQC2:   STFPS   @(R0)+            ;TEST INSTRUCTION.
115366  020027  115434                  CMP     R0,#QQCTB2+2      ;IS R0 CORRECT?
115372  001021                          BNE     QQC10             ;BRANCH IF NOT CORRECT.
115374  023727  115422  106653          CMP     @#QQCTB1,#106653  ;IS THE RESULT CORRECT?
115402  001015                          BNE     QQC10             ;BRANCH IF NOT CORRECT.
115404  023727  115432  115422          CMP     @#QQCTB2,#QQCTB1  ;IS THE RESULT CORRECT?
115412  001011                          BNE     QQC10             ;BRANCH IF NOT CORRECT.
115414  000411                          BR      QQCDONE

                                ;TEST DATA BUFFER:
115416  177777  177777          QQCTB0: .WORD   -1,-1
115422  177777  177777  177777  QQCTB1: .WORD   -1,-1,-1,-1
115430  177777
115432  177777  177777          QQCTB2: .WORD   -1,-1
115436                          QQC10:
115436  104000                          EMT                               ;
115440                          QQCDONE:
115440  004767  003030                  JSR     PC,.RSET          ;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?).



                                        ;*************************************************************************
                                        ;TEST 564       DESTINATION MODES, MODE 5 (FL=0), TEST
                                        ;*************************************************************************
 115444                                 TS564:


 115444  012700  115542                         MOV     #RRCTB0,R0       ;SET UP THE DATA BUFFER.
 115450  012701  000006                         MOV     #6,R1
 115454  012720  177777                 1$:     MOV     #-1,(R0)+
 115460  077103                                 SOB     R1,1$
 115462  012700  004301                         MOV     #004301,R0
 115466  012737  115562  000004                 MOV     #RRC10,@#ERRVECT  ;SET UP FOR TRAPS TO VECTOR 4.
 115474  170100                                 LDFPS   R0               ;SET UP FPS.
 115476  012700  115560                         MOV     #RRCTB2+2,R0
 115502  012760  115546  177776                 MOV     #RRCTB1,-2(R0)

 115510  170250                         RRC2:   STFPS   @-(R0)           ;TEST INSTRUCTION.
 115512  020027  115556                         CMP     R0,#RRCTB2       ;IS R0 CORRECT?
 115516  001021                                 BNE     RRC10            ;BRANCH IF NOT CORRECT.
 115520  023727  115546  004301                 CMP     @#RRCTB1,#004301  ;IS THE RESULT CORRECT?
 115526  001015                                 BNE     RRC10            ;BRANCH IF NOT CORRECT.
 115530  023727  115556  115546                 CMP     @#RRCTB2,#RRCTB1  ;IS THE RESULT CORRECT?
 115536  001011                                 BNE     RRC10            ;BRANCH IF NOT CORRECT.
 115540  000411                                 BR      RRCDONE

                                        ;TEST DATA BUFFER:
 115542  177777  177777                 RRCTB0: .WORD   -1,-1
 115546  177777  177777  177777         RRCTB1: .WORD   -1,-1,-1,-1
 115554  177777
 115556  177777  177777                 RRCTB2: .WORD   -1,-1
 115562                                 RRC10:
 115562  104000                                 EMT                      ;
 115564                                 RRCDONE:
 115564  004767  002704                         JSR     PC,.RSET         ;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?).



                                        ;*************************************************************************
                                        ;TEST 565       DESTINATION MODES, MODE 6 (FL=0), TEST
                                        ;*************************************************************************
 115570                                 TS565:


 115570  012700  115672                         MOV     #SSCTB0,R0       ;SET UP THE DATA BUFFER.
 115574  012701  000006                         MOV     #6,R1
 115600  012720  177777                 1$:     MOV     #-1,(R0)+
 115604  077103                                 SOB     R1,1$
 115606  012700  102514                         MOV     #102514,R0
 115612  012737  115706  000004                 MOV     #SSC10,@#ERRVECT  ;SET UP FOR TRAPS TO VECTOR 4.
```

```
115620  170100                          LDFPS   R0                ;SET UP FPS.
115622  005001                          CLR     R1
115624  012700  110475                  MOV     #SSCTB1-5201,R0

115630  170260  005201          SSC2:   STFPS   5201(R0)               ;TEST INSTRUCTION.
115634  020127  000000                  CMP     R1,#0             ;WAS PC CORRECT AFTER EXECUTION?
115640  001022                          BNE     SSC10             ;BRANCH IF NOT CORRECT.
115642  020027  110475                  CMP     R0,#SSCTB1-5201 ;IS R0 CORRECT?
115646  001017                          BNE     SSC10             ;BRANCH IF NOT CORRECT.
115650  023727  115676  102514          CMP     @#SSCTB1,#102514  ;IS THE RESULT CORRECT?
115656  001013                          BNE     SSC10             ;BRANCH IF NOT CORRECT.
115660  023727  115700  177777          CMP     @#SSCTB1+2,#-1  ;IS THE RESULT CORRECT?
115666  001007                          BNE     SSC10             ;BRANCH IF NOT CORRECT.
115670  000407                          BR      SSCDONE

                                ;TEST DATA BUFFER:
115672  177777  177777          SSCTB0: .WORD   -1,-1
115676  177777  177777  177777  SSCTB1: .WORD   -1,-1,-1,-1
115704  177777
115706                          SSC10:
115706  104000                          EMT                       ;
115710                          SSCDONE:
115710  004767  002560                  JSR     PC,.RSET          ;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?).



                                ;*************************************************************************
                                ;TEST 566       DESTINATION MODES, MODE 7 (FL=0), TEST
                                ;*************************************************************************
115714                          TS566:

115714  012700  116024                  MOV     #TTCTB0,R0        ;SET UP THE DATA BUFFER.
115720  012701  000010                  MOV     #10,R1
115724  012720  177777          1$:     MOV     #-1,(R0)+
115730  077103                          SOB     R1,1$
115732  012700  103747                  MOV     #103747,R0
115736  012737  116044  000004          MOV     #TTC10,@#ERRVECT  ;SET UP FOR TRAPS TO VECTOR 4.
115744  170100                          LDFPS   R0                ;SET UP FPS.
115746  005001                          CLR     R1
115750  012700  110637                  MOV     #TTCTB2-5201,R0
115754  012760  116030  005201          MOV     #TTCTB1,5201(R0)

115762  170270  005201          TTC2:   STFPS   @5201(R0)              ;TEST INSTRUCTION.
115766  022701  000000                  CMP     #0,R1             ;WAS PC CORRECT AFTER EXECUTION?
115772  001024                          BNE     TTC10             ;BRANCH IF NOT CORRECT.
115774  020027  110637                  CMP     R0,#TTCTB2-5201 ;IS R0 CORRECT?
116000  001021                          BNE     TTC10             ;BRANCH IF NOT CORRECT.
116002  023727  116030  103747          CMP     @#TTCTB1,#103747  ;IS THE RESULT CORRECT?
116010  001015                          BNE     TTC10             ;BRANCH IF NOT CORRECT.
116012  023727  116032  177777          CMP     @#TTCTB1+2,#-1  ;IS THE RESULT CORRECT?
116020  001011                          BNE     TTC10             ;BRANCH IF NOT CORRECT.
116022  000411                          BR      TTCDONE
```

```
                                        ;TEST DATA BUFFER:
116024  177777  177777                  TTCTB0: .WORD   -1,-1
116030  177777  177777  177777          TTCTB1: .WORD   -1,-1,-1,-1
116036  177777
116040  177777  177777                  TTCTB2: .WORD   -1,-1
116044                                  TTC10:
116044  104000                                  EMT                     ;
116046                                  TTCDONE:
116046  004767  002422                          JSR     PC,.RSET        ;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?).

                                        ;*************************************************************************
                                        ;TEST 567       DESTINATION MODES, MODE 2 (FL=1), TEST
                                        ;*************************************************************************
116052                                  TS567:
116052  012700  000300                          MOV     #300,R0         ;SET UP FPS.
116056  170100                                  LDFPS   R0
116060  012700  116104                          MOV     #UUCTP1,R0      ;SET UP THE AC0 OPERAND.
116064  172410                                  LDD     (R0),AC0
116066  012700  116116                          MOV     #UUCBF0,R0

116072  175420                          UUC2:   STCDL   AC0,(R0)+       ;TEST INSTRUCTION.

116074  020027  116122                          CMP     R0,#UUCBF0+4    ;IS R0 CORRECT?
116100  001411                                  BEQ     UUCDONE
116102  104000                                  EMT                     ;
                                        ;TEST DATA BUFFER:
116104  000000  000000  000000          UUCTP1: .WORD   0,0,0,0
116112  000000
116114  177777                                  -1
116116  177777  177777  177777          UUCBF0: .WORD   -1,-1,-1

116124                                  UUCDONE:
116124  004767  002344                          JSR     PC,.RSET        ;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?).

                                        ;*************************************************************************
                                        ;TEST 570       DESTINATION MODES, MODE 4 (FL=1), TEST
                                        ;*************************************************************************
116130                                  TS570:

116130  012700  000300                          MOV     #300,R0         ;SET UP FPS.
116134  170100                                  LDFPS   R0
116136  012700  116162                          MOV     #VVCTP1,R0      ;SET UP THE AC0 OPERAND.
116142  172410                                  LDD     (R0),AC0
116144  012700  116200                          MOV     #VVCBF0+4,R0

116150  175440                          VVC2:   STCDL   AC0,-(R0)               ;TEST INSTRUCTION.

116152  020027  116174                          CMP     R0,#VVCBF0      ;IS R0 CORRECT?
```

```
116156  001411                          BEQ     VVCDONE
116160  104000                          EMT                             ;
                                ;TEST DATA BUFFER:
116162  000000  000000  000000  VVCTP1: .WORD   0,0,0,0
116170  000000
116172  177777                          -1
116174  177777  177777  177777  VVCBFO: .WORD   -1,-1,-1

116202                          VVCDONE:
116202  004767  002266                  JSR     PC,.RSET        ;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?).

                                ;*******************************************************************************
                                ;TEST 571       STCDI AND STCDL TEST
                                ;*******************************************************************************
116206                          TS571:

                                ;FIRST TEST STC WITH EXP=100 (EXCESS 200)
116206  004737  117144          WWC1:   JSR     PC,@#STCSUB     ;GO EXECUTE THE INSTRUCTION.
116212  020000  000000  000000  1$:     .WORD   20000,0,0,0     ;ACO OPERAND.
116220  000000
116222  000000  000000          2$:     .WORD   0,0             ;EXPECTED RESULT.
116226  177777  177777          3$:     .WORD   -1,-1           ;ERROR RES.
116232  040300                  4$:     40300                   ;FPS BEFORE EXECUTION.
116234  040304                          40304                   ;FPS AFTER EXECUTION.
116236  140304                          140304                  ;ANTICIPATED ERRONEOUS FPS.
116240  177777                          -1              ;REPORT RESULT INCORRECT.
                                ;EXP=0 (OCT)    FL=1    FIC=0
116242  004737  117144          WWC2:   JSR     PC,@#STCSUB     ;GO EXECUTE THE INSTRUCTION.
116246  040000  000000  000000  1$:     .WORD   40000,0,0,0     ;AC     ;ACO OPERAND.
116254  000000
116256  000000  000000          2$:     .WORD   0,0             ;EXPECTED RESULT.
116262  177777  177777          3$:     .WORD   -1,-1           ;ANTICIPATED ERRONEOUS RESULT.
116266  040313                  4$:     40313                   ;FPS BEFORE EXECUTION.
116270  040304                          40304                   ;FPS AFTER EXECUTION.
116272  140304                          140304                  ;ANTICIPATED ERRONEOUS FPS.
116274  177777                          -1              ;EXPECTED FEC.
                                ;EXP=37 (OCT)   FL=1    FIC=1
116276  004737  117144          WWC3:   JSR     PC,@#STCSUB     ;GO EXECUTE THE INSTRUCTION.
116302  047667  075757  157737  1$:     .WORD   47667,75757,157737,167773       ;ACO OPERAND.
116310  167773
116312  055675  173757          2$:     .WORD   55675,173757            ;EXPECTED RESULT.
116316  122102  004021          3$:     .WORD   122102,004021           ;ANTICIPATED ERRONEOUS RESULT.
116322  040717                  4$:     40717                   ;FPS BEFORE EXECUTION.
116324  040700                          40700                   ;FPS AFTER EXECUTION.
116326  140705                          140705                  ;ANTICIPATED ERRONEOUS FPS.
116330  177777                          -1              ;EXPECTED FEC.
                                ;EXP=40 (OCT)   FL=1    FIC=1
116332  004737  117144          WWC4:   JSR     PC,@#STCSUB     ;GO EXECUTE THE INSTRUCTION.
116336  050000  000000  000000  1$:     .WORD   50000,0,0,0     ;ACO OPERAND.
116344  000000
116346  000000  000000          2$:     .WORD   0,0             ;EXPECTED RESULT.
116352  177777  177777          3$:     .WORD   -1,-1           ;ANTICIPATED ERRONEOUS RESULT.
```

```
116356 040700                      4$:     40700                      ;FPS BEFORE EXECUTION.
116360 140705                              140705                     ;FPS AFTER EXECUTION.
116362 040705                              40705                      ;ANTICIPATED ERRONEOUS FPS.
116364 000006                              6                  ;EXPECTED FEC.

                                   ;EXP=40 (OCT)   FL=1   FIC=0
116366 004737 117144               WWC5:   JSR     PC,@#STCSUB        ;GO EXECUTE THE INSTRUCTION.
116372 050000 000000 000000        1$:     .WORD   50000,0,0,0        ;ACO OPERAND.
116400 000000
116402 000000 000000               2$:     .WORD   0,0                ;EXPECTED RESULT.
116406 177777 177777               3$:     .WORD   -1,-1              ;ANTICIPATED ERRONEOUS RESULT.
116412 040312                      4$:     40312                      ;FPS BEFORE EXECUTION.
116414 040305                              40305                      ;FPS AFTER EXECUTION.
116416 140305                              140305                     ;ANTICIPATED ERRONEOUS FPS.
116420 177777                              -1                 ;EXPECTED FEC.
                                   ;EXP=30 (OCT)   FL=1   FIC=1
116422 004737 117144               WWC6:   JSR     PC,@#STCSUB        ;GO EXECUTE THE INSTRUCTION.
116426 046000 000001 000000        1$:     .WORD   46000,1,0,0        ;ACO OPERAND.
116434 000000
116436 000200 000001               2$:     .WORD   200,1              ;EXPECTED RESULT.
116442 177777 177777               3$:     .WORD   -1,-1              ;ANTICIPATED ERRONEOUS RESULT.
116446 040700                      4$:     40700                      ;FPS BEFORE EXECUTION.
116450 040700                              40700                      ;FPS AFTER EXECUTION.
116452 177777                              -1                         ;ANTICIPATED ERRONEOUS FPS.
116454 177777                              -1                 ;EXPECTED FEC.
                                   ;EXP=27 (OCT)   FL=1   FIC=1
116456 004737 11,144               WWC7:   JSR     PC,@#STCSUB        ;GO EXECUTE THE INSTRUCTION.
116462 045600 000001 000000        1$:     .WORD   45600,1,0,0        ;ACO OPERAND.
116470 000000
116472 000100 000000               2$:     .WORD   100,0              ;EXPECTED RESULT.
116476 177777 177777               3$:     .WORD   -1,-1              ;ANTICIPATED ERRONEOUS RESULT.
116502 040707                      4$:     40707                      ;FPS BEFORE EXECUTION.
116504 040700                              40700                      ;FPS AFTER EXECUTION.
116506 177777                              -1                         ;ANTICIPATED ERRONEOUS FPS.
116510 177777                              -1                 ;EXPECTED FEC.
                                   ;EXP=17 (OCT)   FL=0   FIC=1
116512 004737 117144               WWC10:  JSR     PC,@#STCSUB        ;GO EXECUTE THE INSTRUCTION.
116516 043600 000000 000000        1$:     .WORD   43600,0,0,0        ;ACO OPERAND.
116524 000000
116526 040000 177777               2$:     .WORD   40000,-1                   ;EXPECTED RESULT.
116532 000000 177777               3$:     .WORD   0,-1               ;ANTICIPATED ERRONEOUS RESULT.
116536 040600                      4$:     40600                      ;FPS BEFORE EXECUTION.
116540 040600                              40600                      ;FPS AFTER EXECUTION.
116542 140604                              140604                     ;ANTICIPATED ERRONEOUS FPS.
116544 177777                              -1                 ;EXPECTED FEC.

                                   ;EXP=20 (OCT)   FL=0   FIC=1
116546 004737 117144               WWC11:  JSR     PC,@#STCSUB        ;GO EXECUTE THE INSTRUCTION.
116552 044000 000000 000000        1$:     .WORD   44000,0,0,0        ;ACO OPERAND.
116560 000000
116562 000000 177777               2$:     .WORD   0,-1               ;EXPECTED RESULT.
116566 177777 177777               3$:     .WORD   -1,-1              ;ANTICIPATED ERRONEOUS RESULT.
116572 040600                      4$:     40600                      ;FPS BEFORE EXECUTION.
116574 140605                              140605                     ;FPS AFTER EXECUTION.
116576 040600                              40600                      ;ANTICIPATED ERRONEOUS FPS.
116600 000006                              6                  ;EXPECTED FEC.
```

```
                                  ;EXP=10 (OCT), AC NEGATIVE, FL=0, FIC=1
116602  004737  117144            WWC12:  JSR     PC,@#STCSUB       ;GO EXECUTE THE INSTRUCTION.
116606  142000  000000  000000    1$:     .WORD   142000,0,0,0      ;ACO OPERAND.
116614  000000
116616  177600  177777            2$:     .WORD   177600,-1                 ;EXPECTED RESULT.
116622  000200  000000            3$:     .WORD   200,0             ;ANTICIPATED ERRONEOUS RESULT.
116626  040600                    4$:     40600                     ;FPS BEFORE EXECUTION.
116630  040610                            40610                     ;FPS AFTER EXECUTION.
116632  040600                            40600                     ;ANTICIPATED ERRONEOUS FPS.
116634  177777                            -1                ;EXPECTED FEC.
                                  ;EXP=37 (OCT), FL=1, FIC=1, AC NEG.
116636  004737  117144            WWC13:  JSR     PC,@#STCSUB       ;GO EXECUTE THE INSTRUCTION.
116642  147600  000000  000000    1$:     .WORD   147600,0,0,0      ;ACO OPERAND.
116650  000000
116652  140000  000000            2$:     .WORD   140000,0                  ;EXPECTED RESULT.
116656  137777  000000            3$:     .WORD   137777,0                  ;ANTICIPATED ERRONEOUS RESULT.
116662  040700                    4$:     40700                     ;FPS BEFORE EXECUTION.
116664  040710                            40710                     ;FPS AFTER EXECUTION
116666  177777                            -1                        ;ANTICIPATED ERRONEOUS FPS.
116670  177777                            -1                ;EXPECTED FEC.
                                  ;EXP=37  (OCT),  FL=1,  FIC=1,  AC  NEG
116672  004737  117144            WWC14:  JSR     PC,@#STCSUB       ;GO EXECUTE THE INSTRUCTION.
116676  147600  000000  001000    1$:     .WORD   147600,0,1000,0   ;ACO OPERAND.
116704  000000
116706  137777  177777            2$:     .WORD   137777,177777             ;EXPECTED RESULT.
116712  140000  177777            3$:     .WORD   140000,177777             ;ANTICIPATED ERRONEOUS RESULT.
116716  040707                    4$:     40707                     ;FPS BEFORE EXECUTION.
116720  040710                            40710                     ;FPS AFTER EXECUTION.
116722  177777                            -1                        ;ANTICIPATED ERRONEOUS FPS.
116724  177777                            -1                ;EXPECTED FEC.
                                  ;EXP=41 (OCT),  AC NEG, FL=1, FIC=1
116726  004737  117144            WWC15:  JSR     PC,@#STCSUB       ;GO EXECUTE THE INSTRUCTION.
116732  150200  000000  000000    1$:     .WORD   150200,0,0,0      ;ACO OPERAND.
116740  000000
116742  000000  000000            2$:     .WORD   0,0               ;EXPECTED RESULT.
116746  177777  177777            3$:     .WORD   -1,-1             ;ANTICIPATED ERRONEOUS RESULT.
116752  040700                    4$:     40700                     ;FPS BEFORE EXECUTION.
116754  140705                            140705                    ;FPS AFTER EXECUTION.
116756  177777                            -1                        ;ANTICIPATED ERRONEOUS FPS.
116760  000006                            6                 ;EXPECTED FEC.
                                  ;EXP=40 (OCT),  AC NEG,  FL=1,  FIC=1
116762  004737  117144            WWC16:  JSR     PC,@#STCSUB       ;GO EXECUTE THE INSTRUCTION.
116766  150000  000001  000000    1$:     .WORD   150000,1,0,0      ;ACO OPERAND.
116774  000000
116776  000000  000000            2$:     .WORD   0,0               ;EXPECTED RESULT.
117002  100000  177600            3$:     .WORD   100000,-200               ;ANTICIPATED ERRONEOUS RESULT.
117006  040700                    4$:     40700                     ;FPS BEFORE EXECUTION.
117010  140705                            140705                    ;FPS AFTER EXECUTION.
117012  040700                            40700                     ;ANTICIPATED ERRONEOUS FPS.
117014  000006                            6                 ;EXPECTED FEC.
                                  ;EXP=40, AC NEGATIVE, FL=1, FIC=1
117016  004737  117144            WWC17:  JSR     PC,@#STCSUB       ;GO EXECUTE THE INSTRUCTION.
117022  150001  000000  000000    1$:     .WORD   150001,0,0,0      ;ACO OPERAND.
117030  000000
117032  000000  000000            2$:     .WORD   0,0               ;EXPECTED RESULT.
117036  077400  000000            3$:     .WORD   77400,0           ;ANTICIPATED ERRONEOUS RESULT.
```

```
117042  040700                          4$:     40700                   ;FPS BEFORE EXECUTION.
117044  140705                                  140705                  ;FPS AFTER EXECUTION.
117046  177777                                  -1                      ;ANTICIPATED ERRONEOUS FPS.
117050  000006                                  6               ;EXPECTED FEC.
                                        ;EXP 40   (OCT), AC MOST NEG LONG INT, FL=1
                                        ;FIC=1
117052  004737  117144                  WWC20:  JSR     PC,@#STCSUB     ;GO EXECUTE THE INSTRUCTION.
117056  150000  000000  000000          1$:     .WORD   150000,0,0,0    ;ACO OPERAND.
117064  000000
117066  100000  000000                  2$:     .WORD   100000,0                ;EXPECTED RESULT.
117072  000000  000000                  3$:     .WORD   0,0             ;ANTICIPATED ERRONEOUS RESULT.
117076  040700                          4$:     40700                           ;FPS BEFORE EXECUTION.
117100  040710                                  40710                   ;FPS AFTER EXECUTION.
117102  140705                                  140705                  ;ANTICIPATED ERRONEOUS FPS.
117104  177777                                  -1              ;EXPECTED FEC.
                                        ;EXP=20, AC = MOST NEG INTEGER, FL=0, FIC=1

117106  004737  117144                  WWC21:  JSR     PC,@#STCSUB     ;GO EXECUTE THE INSTPUCTION.
117112  144000  000001  000000          1$:     .WORD   144000,1,0,0    ;ACO OPERAND.
117120  000000
117122  100000  177777                  2$:     .WORD   100000,-1               ;EXPECTED RESULT.
117126  100000  177400                  3$:     .WORD   100000,177400           ;ANTICIPATED ERRONEOUS RESULT.
117132  040600                          4$:     40600                   ;FPS BEFORE EXECUTION.
117134  040610                                  40610                   ;FPS AFTER EXECUTION.
117136  140605                                  140605                  ;ANTICIPATED ERRONEOUS FPS.
117140  177777                                  -1              ;EXPECTED FEC.
117142  000457                          6$:     BR      WWCDONE

                                        ;THIS SUBROUTINE, STCSUB, IS USED TO SET UP THE OPERANDS, EXECUTE
                                        ;THE STCDI OR STCDL INSTRUCTION AND CHECK THE RESULTS. A CALL
                                        ;TO IT IS MADE THUS:
                                        ;
                                        ;               JSR     PC,@#STCSUB
                                        ;               ACARG:  .WORD   X,X,X,X         ;AC OPERAND
                                        ;               RES:    .WORD   X,X             ;EXPECTED RESULT
                                        ;               ERRES:  .WORD   X,X             ;ERROR RESULT
                                        ;               FPSB:   .WORD   X               ;FPS BEFORE EXECUTION
                                        ;               FPSA:   .WORD   X               ;FPS AFTER EXECUTION
                                        ;               ERFPS:  .WORD   X               ;ERROR FPS.
                                        ;               FEC:    .WORD   X               ;EXPECTED FEC
                                        ;               ERR1:   ERROR   X               ;DATA ERROR.
                                        ;                       BR      CONT
                                        ;               ERR2:   ERROR   X               ;FPS ERROR.
                                        ;               CONT:                           ;RETURN ADDRESS
                                        ;
                                        ;THE OPERANDS ARE SET UP (USING ACO AS THE ACCUMULATOR). THEN
                                        ;THE STCDI OR STCDL INSTRUCTION IS EXECUTED.
                                        ;THE RESULT IS CHECKED AGAINST RES. IF THE RESULT IS CORRECT THEN THE FPS IS
                                        ;COMPARED WITH FPSA IF THIS TOO IS CORRECT STCSUB RETURNS CONTROL
                                        ;TO THE CALLING ROUTINE AT CONT. IF THE FPS IS BAD STCSUB
                                        ;COMPARE IT TO ERROR FPS. IF THIS MATCHES THEN STCSUB WILL RETURN
                                        ;TO THE ERROR CALL AT ERR2, OTHERWISE STCSUB ITSELF
                                        ;REPORTS THIS FAILURE AND THEN RETURNS TO CONT. IF THE RESULT OF THE
                                        ;STCDI OR STCDL IS INCORRECT, THE INCORRECT RESULT IS COMPARED WITH THE
                                        ;ANTICIPATED FAILING DATA PATTERN, ERRES. IF THE FAILURE IN
                                        ;THE RESULT WAS ANTICIPATED CORRECTLY TO BE ERRES THEN STCSUB
```

```
                                  ;WILL TRANSFER CONTROL TO THE ERROR CALL AT ERR1. OTHERWISE THE
                                  ;RESULT WAS INCORRECT BUT WAS NOT ANTICIPATED AND STCSUB WILL
                                  ;REPORT THE FAILURE AFTER WHICH CONTROL WILL BE PASSED TO CONT.

117144  012601                    STCSUB: MOV     (SP)+,R1        ;GET A POINTER TO THE ARGUMENTS.
117146  012700  000200                    MOV     #200,R0         ;SET UP THE AC0 OPERAND.
117152  170100                            LDFPS   R0
117154  010100                            MOV     R1,R0
117156  172410                            LDD     (R0),AC0
117160  012702  117272                    MOV     #STCIBF,R2      ;INITIALIZE THE OUT PUT BUFFER.
117164  012700  000004                    MOV     #4,R0
117170  012722  177777            1$:     MOV     #-1,(R2)+
117174  077003                            SOB     R0,1$
117176  016100  000020                    MOV     20(R1),R0       ;SET THE FPS.
117202  170100                            LDFPS   R0
117204  012700  117272                    MOV     #STCIBF,R0
117210  175410                    2$:     STCDL   AC0,(R0)        ;TEST INSTRUCTION.

117212  170204                            STFPS   R4              ;GET THE FPS.
117214  170305                            STST    R5              ;GET THE FEC.
117216  010102                            MOV     R1,R2
117220  062702  000010                    ADD     #10,R2
117224  012700  117272                    MOV     #STCIBF,R0      ;SEE IF THE RESULT IS CORRECT.
117230  012703  000002                    MOV     #2,R3
117234  022022                    3$:     CMP     (R0)+,(R2)+
117236  001014                            BNE     10$
117240  077303                            SOB     R3,3$
117242  016102  000022                    MOV     22(R1),R2
117246  020204                            CMP     R2,R4           ;SEE IF THE FPS IS CORRECT.
117250  001007                            BNE     10$             ;BRANCH IF INCORRECT.
117252  005702                            TST     R2
117254  100003                            BPL     4$
117256  026105  000026                    CMP     26(R1),R5       ;SEE IF THE FEC IS CORRECT.
117262  001002                            BNE     10$             ;BRANCH IF INCORRECT.

117264  000161  000030            4$:     JMP     30(R1)          ;RETURN.
117270                            10$:
117270  104000                            EMT                     ;

                                  ;DATA BUFFER:
117272  177777  177777  177777    STCIBF: .WORD   -1,-1,-1,-1
117300  177777

117302                            WWCDONE:
117302  004767  001166                    JSR     PC,.RSET        ;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?).


                                  ;*************************************************************************
                                  ;TEST 572        STCFL AND STCFI TEST
                                  ;*************************************************************************
117306                            TS572:

```

```
                                          ;EXPONENT=37, FL=1
117306  004737  117144                           JSR     PC,@#STCSUB      ;GO EXECUTE THE INSTRUCTION.
117312  047777  177777  177777    1$:            .WORD   47777,-1,-1,-1   ;ACO OPERAND.
117320  177777
117322  077777  177600            2$:            .WORD   77777,177600             ;EXPECTED RESULT.
117326  077777  177777            3$:            .WORD   77777,177777             ;ANTICIPATED ERRONEOUS RESULT.
117332  040100                    4$:            40100                    ;FPS BEFORE EXECUTION.
117334  040100                                   40100                    ;FPS AFTER EXECUTION.
117336  177777                                   -1                       ;ANTICIPATED ERRONEOUS FPS.
117340  177777                                   -1              ;EXPECTED FEC.
117342                            XXCDONE:
117342  004767  001126                           JSR     PC,.RSET         ;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?).



                                  ;*******************************************************************************
                                  ;TEST 573       STEXP TEST
                                  ;*******************************************************************************
117346                            TS573:

                                  ; EXP = 100 (EXCESS 200)
117346  004737  117554            YYC1:   JSR     PC,@#STXSUB
117352  020000  000000  000000    1$:     .WORD   20000,0,0,0                               ;AC
117360  000000
117362  177700                    2$:     -100                                              ;EXP RES
117364  052525                    3$:     52525                                             ;ERROR EXP.
117366  040000                    4$:     40000                                             ;FPSB
117370  040010                            40010                                             ;FPSA
117372  040000                            40000                                             ;ERROP FPS
                                  ; EXP = 200 (EXCESS 200)
117374  004737  117554            YYC2:   JSR     PC,@#STXSUB      ;GO EXECUTE THE INSTRUCTION.
117400  040000  000000  000000    1$:     .WORD   40000,0,0,0      ;ACO OPERAND.
117406  000000
117410  000000                    2$:     0                        ;EXPECTED EXPONENT RESULT.
117412  052525                    3$:     52525           ;ANTICIPATED ERRONEOUS RESULT.
117414  040000                    4$:     40000                    ;FPS BEFORE EXECUTION.
117416  040004                            40004                    ;FPS AFTER EXECUTION.
117420  040000                            40000                    ;ANTICIPATED ERRONEOUS FPS.
                                  ; EXP = 201 (EXCESS 200)
117422  004737  117554            YYC3:   JSR     PC,@#STXSUB      ;GO EXECUTE THE INSTRUCTION.
117426  040200  000000  000000    1$:     .WORD   40200,0,0,0      ;ACO OPERAND.
117434  000000
117436  000001                    2$:     1                        ;EXPECTED EXPONENT RESULT.
117440  052525                    3$:     52525           ;ANTICIPATED ERRONEOUS RESULT.
117442  040000                    4$:     40000                    ;FPS BEFORE EXECUTION.
117444  040000                            40000                    ;FPS AFTER EXECUTION.
117446  040004                            40004                    ;ANTICIPATED ERRONEOUS FPS.
                                  ; EXP = 375  (EXCESS 200)

117450  004737  117554            YYC4:   JSR     PC,@#STXSUB      ;GO EXECUTE THE INSTRUCTION.
117454  077200  000000  000000    1$:     .WORD   77200,0,0,0      ;ACO OPERAND.
117462  000000
```

```
117464  000175                          2$:     175                     ;EXPECTED EXPONENT RESULT.
117466  052525                          3$:     52525           ;ANTICIPATED ERRONEOUS RESULT.
117470  040000                          4$:     40000                   ;FPS BEFORE EXECUTION.
117472  040000                                  40000                   ;FPS AFTER EXECUTION.
117474  040010                                  40010                   ;ANTICIPATED ERRONEOUS FPS.
                                        ; EXP = 1  (EXCESS 200)

117476  004737  117554                  YYC5:   JSR     PC,@#STXSUB     ;GO EXECUTE THE INSTRUCTION.
117502  000200  000000  000000          1$:     .WORD   200,0,0,0       ;AC0 OPERAND.
117510  000000
117512  177601                          2$:     -177                    ;EXPECTED EXPONENT RESULT.
117514  052525                          3$:     52525           ;ANTICIPATED ERRONEOUS RESULT.
117516  040000                          4$:     40000                   ;FPS BEFORE EXECUTION.
117520  040010                                  40010                   ;FPS AFTER EXECUTION.
117522  040000                                  40000                   ;ANTICIPATED ERRONEOUS FPS.
                                        ; EXP = 156  (EXCESS 200)

117524  004737  117554                  YYC6:   JSR     PC,@#STXSUB     ;GO EXECUTE THE INSTPUCTION.
117530  033400  000000  000000          1$:     .WORD   33400,0,0,0     ;AC0 OPERAND.
117536  000000
117540  177756                          2$:     -22                     ;EXPECTED EXPGNENT RESULT.
117542  052525                          3$:     52525           ;ANTICIPATED ERRONEOUS RESULT.
117544  047707                          4$:     47707                   ;FPS BEFORE EXECUTION.
117546  047710                                  47710                   ;FPS AFTER EXECUTION.
117550  177777                                  -1                      ;ANTICIPATED ERRONEOUS FPS.
117552  000452                                  BR      YYCDONE

                                        ;THIS SUBROUTINE, STXSUB, IS USED TO SET UP THE OPERANDS, EXECUTE
                                        ;THE STEXP INSTRUCTION AND CHECK THE RESULTS. A CALL
                                        ;TO IT IS MADE THUS:
                                        ;
                                        ;               JSR     PC,@#STXSUB
                                        ;               ACARG:  .WORD   X,X,X,X         ;AC OPERAND
                                        ;               RES:    .WORD   X               ;EXPECTED RESULT
                                        ;               ERRES:  .WORD   X               ;ERROR RESULT
                                        ;               FPSB:   .WORD   X               ;FPS BEFORE EXECUTION
                                        ;               FPSA:   .WORD   X               ;FPS AFTER EXECUTION
                                        ;               ERFPS:  .WORD   X               ;ERROR FPS.
                                        ;               ERR1:   ERROR   X               ;DATA ERROR.
                                        ;                       BR      CONT
                                        ;               ERR2:   ERROR   X               ;FPS ERROR.
                                        ;               CONT:                           ;RETURN ADDRESS
                                        ;
                                        ;THE OPERANDS ARE SET UP (USING AC0 AS THE ACCUMULATOR). THEN
                                        ;THE STEXP INSTRUCTION IS EXECUTED.
                                        ;THE RESULT IS CHECKED AGAINST RES. IF THE RESULT IS CORRECT THEN THE FPS IS
                                        ;COMPARED WITH FPSA IF THIS TOO IS CORRECT STXSUB RETURNS CONTROL
                                        ;TO THE CALLING ROUTINE AT CONT. IF THE FPS IS BAD STXSUB
                                        ;COMPARE IT TO ERROR FPS. IF THIS MATCHES THEN STXSUB WILL RETURN
                                        ;TO THE ERROR CALL AT ERR2, OTHERWISE STXSUB ITSELF
                                        ;REPORTS THIS FAILURE AND THEN RETURNS TO CONT. IF THE RESULT OF THE
                                        ;STEXP IS INCORRECT, THE INCORRECT RESULT IS COMPARED WITH THE
                                        ;ANTICIPATED FAILING DATA PATTERN, ERRES. IF THE FAILURE IN
                                        ;THE RESULT WAS ANTICIPATED CORRECTLY TO BE ERRES THEN STXSUB
                                        ;WILL TRANSFER CONTROL TO THE ERROR CALL AT ERR1. OTHERWISE THE
                                        ;RESULT WAS INCORRECT BUT WAS NOT ANTICIPATED AND STXSUB WILL
```

```
                                         ;REPORT THE FAILURE AFTER WHICH CONTROL WILL BE PASSED TO CONT.

117554  012601                           STXSUB: MOV     (SP)+,R1           ;GET A POINTER TO THE ARGUMENTS.
117556  010102                                   MOV     R1,R2
117560  012737  123456  117666                   MOV     #123456,@#STXBF
117566  012737  076543  117670                   MOV     #76543,@#STXBF+2
117574  012700  000200                           MOV     #200,R0
117600  170100                                   LDFPS   R0
117602  010100                                   MOV     R1,R0              ;SET UP THE AC0 OPERAND.
117604  172410                                   LDD     (R0),AC0
117606  016100  000016                           MOV     16(R1),R0          ;SET THE FPS.
117612  170100                                   LDFPS   R0
117614  012700  117666                           MOV     #STXBF,R0
117620  175010                           1$:     STEXP   AC0,(R0)           ;TEST INSTRUCTION.
117622  170204                                   STFPS   R4                 ;GET FPS.
117624  026137  000010  117666                   CMP     10(R1),@#STXBF     ;WAS RESULT CORRECT?
117632  001401                                   BEQ     5$
117634  104000                                   EMT                        ;
117636  020461  000016                   5$:     CMP     R4,16(R1)          ;SEE IF THE FPS IS CORRECT.
117642  001401                                   BEQ     10$
117644  104000                                   EMT                        ;
                                         ;SEE IF MORE THAN ONE WORD WAS WRITTEN IN THE OUTPUT BUFFER.
117646  022737  076543  117670          10$:     CMP     #76543,@#STXBF+2
117654  001401                                   BEQ     4$
117656  104000                                   EMT                        ;
117660  000161  000022                   4$:     JMP     22(R1)

117664  177777                                   -1
117666  177777  177777  177777          STXBF:   .WORD   -1,-1,-1,-1,-1
117674  177777  177777

117700                                   YYCDONE:
117700  004767  000570                           JSR     PC,.RSET           ;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?).

                                         ;************************************************************************
                                         ;TEST 574        STST TEST
                                         ;************************************************************************
117704                                   TS574:

117704  012700  040000                           MOV     #40000,R0          ;SET FPS. FID=1.
117710  170100                                   LDFPS   R0

117712  170003                           ZZC2:   .WORD   170003                   ;ILLEGAL FPP
                                                                                  ;OP CODE
117714  012700  117774                           MOV     #ZZCBF,R0          ;SET UP THE OUTPUT BUFFER.
117720  012710  177777                           MOV     #-1,(R0)
117724  012760  177777  000002                   MOV     #-1,2(R0)
117732  170310                           ZZC3:   STST    (R0)                     ;GET FEC AND
                                                                                  ;FEA
117734  170204                                   STFPS   R4                 ;GET FPS.
117736  012700  117774                           MOV     #ZZCBF,R0
117742  022710  000002                           CMP     #2,(R0) ;SEE IF FEC IS CORRECT.
```

```
117746  001010                                  BNE     ZZC10           ;BRANCH IF INCORRECT.
117750  022760  117712  000002                  CMP     #ZZC2,2(R0)     ;SEE IF FEA, ADDRESS, IS CORRECT.
117756  001004                                  BNE     ZZC10           ;BRANCH IF INCORRECT.
117760  022704  140000                          CMP     #140000,R4      ;SEE IF FPS IS CORRECT.
117764  001001                                  BNE     ZZC10           ;BRANCH IF INCORRECT.
117766  000407                                  BR      ZZCDONE
117770                                  ZZC10.
117770  104000                                  EMT                     ;

                                        ;DATA BUFFER:
117772  177777                                  -1
117774  177777  177777  177777          ZZCBF:  .WORD   -1,-1,-1,-1
120002  177777
120004  177777                                  -1

120006                                  ZZCDONE:
120006  004767  000462                          JSR     PC,.RSET        ;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?).


                                        ;**************************************************************************
                                        ;TEST 575       SPECIAL CASE TEST
                                        ;**************************************************************************
120012                                  TS575:
120012  012746  144724                  AAD1:   MOV     #144724, -(SP)  ;PUT FRACTION ON STACK
120016  012746  040600                          MOV     #40600, -(SP)   ;PUT EXPONENT ON STACK
120022  005046                                  CLR     -(SP)           ;PUT SUBTRAHEND FRACTION ON STACK
120024  012746  040600                          MOV     #40600, -(SP)   ;PUT SUBTRAHEND EXPONENT ON STACK
120030  172466  000004                          LDF     4(SP), AC0      ;LOAD FP ACCUMULATORS
120034  173026                                  SUBF    (SP)+, AC0      ;DO SUBTRACTION
120036  174037  120066                          STF     AC0, @#AADBF    ;GET AND STORE ANSWER
120042  022737  036711  120066                  CMP     #36711, @#AADBF ;IS EXPONENT CORRECT
120050  001401                                  BEQ     1$
120052  104000                                  EMT                     ;BAD EXPONENT FROM SUBTRACTION
120054  022737  152000  120070          1$:     CMP     #152000, @#AADBF+2 ;IS FRACTION CORRECT
120062  001403                                  BEQ     AADDONE
120064  104000                                  EMT                     ;FRACTION INCORRECT

120066  000000                          AADBF:  .WORD   0
120070  000000                                  .WORD   0

120072  012706  001000                  AADDONE: MOV    #STBOT, SP      ;RESTORE STACK POINTER
120076  004767  000372                          JSR     PC,.RSET        ;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?).



                                        ;**************************************************************************
                                        ;TEST 576       POWER MONITOR BIT CHECK
                                        ;**************************************************************************
```

```
120102                                  TS576:
120102  013700  000004                          MOV     @#4,R0
120106  013701  000006                          MOV     @#6,R1                  ;SAVE THE TIMEOUT VECTOR
120112  012737  120136  000004                  MOV     #1$,@#4
120120  012737  000340  000006                  MOV     #340,@#6                ;SET UP THE TIMEOUT VECTOR TO WITHIN THE TEST
120126  005737  177766                          TST     @#177766                ;IS THE MAP PRESENT?(NO TRAP TO 1$)
120132  001403                                  BEQ     2$                      ;YES,AND IF EQUAL TO ZERO THEN CLEAN UP
                                                                                ;AND GO TO NEXT TEST
120134  104000                                  EMT                             ;IF POWER HAS FAILED THEN GO REPORT ERROR
120136  005726                          1$:     TST     (SP)+                   ;READJUST THE STACK TO COMPENSATE FOR THE
120140  005726                                  TST     (SP)+                   ;STACK
120142  010037  000004                  2$:     MOV     R0,@#4                  ;RESTORE THE TIMEOUT VECTOR
120146  010137  000006                          MOV     R1,@#6                  ;AND GO TO NEXT TEST

                                        ;*******************************************************************************
                                        ;TEST 577       INTERRUPTABILITY TEST
                                        ;*******************************************************************************
120152                                  TS577:
120152  004567  130250                  BBD1:   JSR     R5,CHKAPT
120156  000520                                  BR      FPEXIT                  ;SKIP TEST IF ON APT AND NOT FIRST PASS

120160  005001                                  CLR     R1                      ;INITIALIZE A COUPLE OF COUNTERS
120162  005000                                  CLR     R0
120164  013767  000064  117170                  MOV     @#64,   $TMP0           ;SAVE INTERRUPT VECTOR
120172  013767  000066  117164                  MOV     @#66,   $TMP1           ;SAVE INTERRUPT PRIORITY
120200  012737  120252  000064                  MOV     #3$,    @#64            ;SET UP INTERRUPT PRIORITY FOR THIS TEST
120206  005037  000066                          CLR     @#66                    ;AND PRIORITY
120212  005067  057560                          CLR     PS                      ;PUT PROCESSOR PRIORITY AT 0
120216  005067  057344                          CLR     TPB                     ;SEND A NULL CHARACTER
120222  105767  057336                  1$:     TSTB    TTCSR                   ;WAIT FOR DONE TO SET
120226  100375                                  BPL     1$
120230  005067  057332                          CLR     TPB                     ;SEND A SECOND CHARACTER
120234  052767  000100  057322                  BIS     #BIT6,  TTCSR           ;SET INTERRUPT ENABLE
120242  005200                          2$:     INC     R0                      ;INCREMENT COUNTER TO GET BASE TIME
120244  001376                                  BNE     2$                      ;CONTINUE LOOPING UNLESS COUNTER GOES TO 0
120246  000005                                  RESET                           ;IF NO INTERRUPT YET KILL IT
120250  104000                                  EMT                             ;NO INTERRUPT OCCURRED IN ALLOTTED TIME
120252  166700  000110                  3$:     SUB     Y,      R0              ;SUBTRACT TIME FOR FP INSTRUCTION
120256  010067  000106                          MOV     R0,     Z               ;SAVE FIRST TIME
120262  012737  120340  000064                  MOV     #7$,    @#64            ;SET UP FOR NEXT INTERRUPT
120270  005100                          4$:     COM     R0                      ;MAKE PRE LOOP COUNTER NEGATIVE
120272  005067  057266                          CLR     TTCSR                   ;MAKE SURE NO INTERRUPT YET
120276  005067  057264                          CLR     TPB                     ;SEND A CHARACTER
120302  105767  057256                  5$:     TSTB    TTCSR                   ;WAIT FOR READY BIT TO SET
120306  100375                                  BPL     5$
120310  005067  057252                          CLR     TPB                     ;SEND SECOND CHARACTER
120314  052767  000100  057242                  BIS     #BIT6,  TTCSR           ;SET INTERRUPT ENABLE
120322  005200                          6$:     INC     R0                      ;DO PRE LOOP
120324  001376                                  BNE     6$
120326  171227  040400                          MULF    #2,     AC2             ;DO FLOATING POINT INSTRUCTION
120332  000240                                  NOP                             ;JUST IN CASE INTERRUPT TAKES TOO LONG
120334  000005                                  RESET                           ;IF NO INTERRUPT CLEAR THE WORLD
120336  104000                                  EMT                             ;INTERRUPT NOT BACK IN ALLOTTED TIME
120340  005201                          7$:     INC     R1                      ;INCREMENT TIMES THROUGH COUNTER
120342  020127  000015                          CMP     R1,     #15             ;HAVE WE PASSED HERE 15 TIMES BEFORE
120346  001411                                  BEQ     BBDDONE                 ;IF YES I MAY NEVER PASS HERE AGAIN
```

```
120350 062767 000002 000012            ADD     #2,      Z          ;IF NO ADD A LITTLE TIME TO PRELOOP
120356 016700 000006                   MOV     Z,       R0         ;PUT NEW COUNT IN COUNTER
120362 000742                          BR      4$                  ;DO IT ALL AGAIN

120364 000000                  X:      .WORD   0
120366 000026                  Y:      .WORD   26
120370 000000                  Z:      .WORD   0

120372 042767 000100 057164    BBDDONE:BIC     #100,    TTCSR      ;CLEAR INTERRUPT ENABLE BEFORE EXITING TEST
120400 016737 116756 000064            MOV     $TMP0,   @#64       ;RESTORE PRINTER VECTOR
120406 016737 116752 000066            MOV     $TMP1,   @#66       ;RESTORE PRINTER PRIORITY
120414 004767 000054                   JSR     PC,.RSET            ;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?).
120420 000167 000134           FPEXIT: JMP     SLU1ST              ;GET OVER SUBROUTINES TO NEXT TEST


120424 012737 120456 000004    ERROR4: MOV     #1$,@#4             ;SET UP THE TIMEOUT VECTOR
120432 012737 000340 000006            MOV     #340,@#6
120440 005737 177766                   TST     @#177766            ;CHECK TO SEE IF THE POWER FAILED,WILL TRAP
                                                                   ;TO 1$ IF UNIBUS MAP BOARD IS NOT PRESENT
                                                                   ;SINCE THE CPUERR(177766)IS ON THE UNIBUS
                                                                   ;MAP BOARD
120444 001404                          BEQ     1$                  ;NO,GO REPORT THE ERROR
120446 012737 000177 001002            MOV     #177,@#$FATAL       ;YES,FLAG THAT THE POWER FAILED
120454 000403                          BR      2$                  ; AND GO SET FATAL ERROR FLAG
120456 012737 000003 001002    1$:     MOV     #3,@#$FATAL                     ;SET UP FATAL ERROR NUMBER
120464 012767 000001 060306    2$:     MOV     #1,$MSGTY                       ;SET FATAL ERROR FLAG
120472 000777                  FPHLT:  BR      .                   ;STAY HERE FOREVER
                                       .SBTTL  FLAG RESET ROUTINE
                               ;*************************************************************************
                               ;*THIS ROUTINE WILL BE CALLED AT THE END OF EACH FLOATING POINT TEST
                               ;*TO RESET THE STACK, CLEAR THE FPS AND REINITIALIZE TRAP VECTORS

120474 012737 120424 000244    .RSET:  MOV     #ERROR4,@#FPVECT
120502 012737 021256 000004            MOV     #T04,@#ERRVECT
120510 012737 021260 000010            MOV     #T010,@#10
120516 011600                          MOV     (SP),R0
120520 012706 001000                   MOV     #STBOT,SP
120524 005004                          CLR     R4
120526 170104                          LDFPS   R4
120530 000110                          JMP     (R0)

                               ;THESE ARE SOME EQUATES USE IN THE LAST THREE SECTIONS
       000001                          BIT0=000001
       000002                          BIT1=000002
       000004                          BIT2=000004
       000010                          BIT3=000010
       000020                          BIT4=000020
       000040                          BIT5=000040
       000100                          BIT6=000100
       000200                          BIT7=000200
       000400                          BIT8=000400
       001000                          BIT9=001000
```

```
                002000                          BIT10=002000
                004000                          BIT11=004000
                010000                          BIT12=010000
                020000                          BIT13=020000
                040000                          BIT14=040000
                100000                          BIT15=100000

        120532  177560                          RCSR:   177560                  ;ADDRESS OF RECEIVER COMMAND/STATUS REGISTER
        120534  177562                          RBUF:   177562                  ;ADDRESS OF RECEIVER BUFFER
        120536  177564                          TCSR:   177564                  ;ADDRESS OF TRANSMITTER COMMAND/STATUS REGISTER
        120540  177566                          TBUF:   177566                  ;ADDRESS OF TRANSMITTER BUFFER
        120542  000060                          RVECT:  60                      ;RECEIVER INTERRUPT VECTOR
        120544  000062                          RPSW:   62
        120546  000064                          TVECT:  64                      ;TRANSMITTER INTERRUPT VECTOR
        120550  000066                          TPSW:   66

                                                ;REAL TIME CLOCK REGISTER AND VECTOR ADDRESSES
        120552  177546                          LKS:    .WORD   177546
        120554  000100                          RTCVT:  .WORD   100
        120556  000102                          RTCPSW: .WORD   102

        120560  032777  000004  100536          SLU1ST: BIT     #4,@SWR
        120566  001402                                  BEQ     1$
        120570  000167  002424                          JMP     KWSTRT
        120574  012737  000004  001004          1$:     MOV     #4,@#$TESTN     ;PUT TEST NUMBER IN MAILBOX
        120602  012737  123150  000030                  MOV     #ERROR5,@#30    ;SET UP FOR CORRECT ERROR CALL



                                                ;*************************************************************************
                                                ;TEST 600       TEST ABILITY TO REFERENCE TCSR
                                                ;*************************************************************************
        120610                                  TS600:
        120610  013703  000004                          MOV     @#4,R3          ;SAVE TIMEOUT VECTOR
        120614  012737  120630  000004                  MOV     #1$,@#4         ;SET UP TIMEOUT VECTOR
        120622  005777  177710                          TST     @TCSR           ;REFERENCE THE XMIT COMMAND/STATUS REG.
        120626  000401                                  BR      4$
        120630                                  1$:
        120630  104000                                  EMT                     ;
        120632  010337  000004                  4$:     MOV     R3,@#4          ;RESTORE TIMEOUT VECTOR



                                                ;*************************************************************************
                                                ;TEST 601       TEST ABILITY TO REFERENCE TBUF
                                                ;*************************************************************************
        120636                                  TS601:
        120636  013703  000004                          MOV     @#4,R3          ;SAVE TIMEOUT VECTOR
        120642  012737  120656  000004                  MOV     #1$,@#4         ;SET UP TIMEOUT VECTOR
        120650  005777  177664                          TST     @TBUF           ;REFERENCE THE XMIT BUFFER
        120654  000401                                  BR      4$
        120656                                  1$:
        120656  104000                                  EMT                     ;
        120660  010337  000004                  4$:     MOV     R3,@#4          ;RESTORE TIMEOUT VECTOR


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

```
                                        ;TEST 602       TEST THAT TCSR BIT7(DONE) CLEARS WHEN XBUF IS LOADED
                                        ;*******************************************************************************
120664                                  TS602:
120664  032737  000001  001020                  BIT     #1,@#$ENV               ;ARE WE RUNNING UNDER APT
120672  001405                                  BEQ     70$                     ;IF NO THEN SERIES OF TESTS
120674  005737  001006                          TST     @#$PASS                 ;IS THIS FIRST PASS
120700  001402                                  BEQ     70$                     ;IF YES THEN DO SERIES OF TESTS
120702  000167  002312                          JMP     KWSTRT                  ;IF NO THEN BYPASS SERIES OF TESTS
120706  005077  177626                  70$:    CLR     @TBUF                   ;LOAD XBUF
120712  105777  177620                          TSTB    @TCSR                   ;CHECK DONE
120716  100006                                  BPL     3$                      ;BR IF CLEAR
                                                                                ;FILL SECOND BUFFER BECUASE REFRESH COULD CAUSE
                                                                                ;FIRST TEST TO FAIL
120720  005077  177614                          CLR     @TBUF                   ;FILL DOUBLE BUFFER
120724  105777  177606                          TSTB    @TCSR                   ;CHECK DONE
120730  100001                                  BPL     3$
120732  104000                                  EMT                             ;
120734  005000                          3$:     CLR     R0                      ;CLEAR TIMER
120736  105777  177574                  4$:     TSTB    @TCSR                   ;CHECK FOR XMIT DONE
120742  100403                                  BMI     5$                      ;IF DONE SETS, BR TO END OF TEST
120744  005200                                  INC     R0                      ;INCREMENT TIMER
120746  001373                                  BNE     4$
120750  104000                                  EMT                             ;
120752                                  5$:


                                        ;*******************************************************************************
                                        ;TEST 603       TEST THAT TCSR "DONE" SETS WITH RESET
                                        ;*******************************************************************************
120752                                  TS603:
120752  005077  177562                          CLR     @TBUF                   ;LOAD TRANSMIT BUFFER
120756  105777  177554                  1$:     TSTB    @TCSR                   ;WAIT FOR DONE
120762  100375                                  BPL     1$
120764  005077  177550                          CLR     @TBUF                   ;LOAD SECOND BUFFER
120770  000240                                  NOP
120772  000005                                  RESET                           ;SET DONE WITH RESET
120774  105777  177536                          TSTB    @TCSR                   ;CHECK FOR DONE SET
121000  100401                                  BMI     TS604
121002  104000                                  EMT                             ;



                                        ;*******************************************************************************
                                        ;TEST 604       TEST ABILITY TO ACCESS RCSR
                                        ;*******************************************************************************
121004                                  TS604:
121004  013703  000004                          MOV     @#4,R3                  ;SAVE TIMEOUT VECTOR
121010  012737  121024  000004                  MOV     #1$,@#4                 ;SET UP TIMEOUT VECTOR
121016  005777  177510                          TST     @RCSR                   ;ACCESS RCSR
121022  000401                                  BR      2$
121024                                  1$:
121024  104000                                  EMT                             ;
121026  010337  000004                  2$:     MOV     R3,@#4                  ;RESTORE TIMEOUT VECTOR


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

```
                                        ;TEST 605         TEST ABILITY TO ACCESS RBUF
                                        ;*************************************************************************
121032                                  TS605:
121032  013703  000004                          MOV     @#4,R3          ;SAVE TIMEOUT VECTOR
121036  012737  121052  000004                  MOV     #1$,@#4         ;SET UP TIMEOUT VECTOR
121044  005777  177464                          TST     @RBUF           ;ACCESS RBUF
121050  000401                                  BR      2$
121052                                  1$:
121052  104000                                  EMT                     ;
121054  010337  000004                  2$:     MOV     R3,@#4          ;RESTORE TIMEOUT VECTOR






                                        ;*************************************************************************
                                        ;TEST 606         TEST THAT BIT2(MAINT. BIT) CAN BE SET & RESET
                                        ;*************************************************************************
121060                                  TS606:
121060  032777  000004  177450                  BIT     #BIT2,@TCSR     ;TEST FOR BIT2 OF TCSR CLEAR
121066  001401                                  BEQ     3$
121070  104000                                  EMT                     ;
121072  052777  000004  177436          3$:     BIS     #BIT2,@TCSR     ;SET BIT2 OF TCSR
121100  032777  000004  177430                  BIT     #BIT2,@TCSR     ;TEST FOR BIT2 SET
121106  001001                                  BNE     4$
121110  104000                                  EMT                     ;
121112  042777  000004  177416          4$:     BIC     #BIT2,@TCSR     ;CLEAR BIT2 OF TCSR
121120  032777  000004  177410                  BIT     #BIT2,@TCSR     ;TEST BIT2 CLEAR
121126  001401                                  BEQ     7$
121130  104000                                  EMT                     ;
121132  052777  000004  177376          7$:     BIS     #BIT2,@TCSR     ;SET BIT2 OF TCSR
121140  000005                                  RESET                   ;CLEAR BIT2 WITH RESET
121142  032777  000004  177366                  BIT     #BIT2,@TCSR     ;TEST FOR BIT2 CLEAR
121150  001401                                  BEQ     TS607
121152  104000                                  EMT                     ;


                                        ;*************************************************************************
                                        ;TEST 607         TEST THAT BIT6(XMIT INT EN) CAN BE SET & RESET
                                        ;*************************************************************************
121154                                  TS607:
121154  017703  177366                          MOV     @TVECT,R3       ;SAVE XMIT VECTOR
121160  012777  121202  177360                  MOV     #1$,@TVECT      ;SET UP INTERRUPT VECTOR FOR ERROR REPORT
121166  106427  000340                          MTPS    #340            ;SET PSW TO PRIORITY 7
121172  032777  000100  177336                  BIT     #BIT6,@TCSR     ;TEST BIT6 OF TCSR
121200  001401                                  BEQ     2$
121202                                  1$:
121202  104000                                  EMT                     ;
121204  052777  000100  177324          2$:     BIS     #BIT6,@TCSR     ;SET BIT6 OF TCSR
121212  032777  000100  177316                  BIT     #BIT6,@TCSR     ;TEST BIT6 OF TCSR
121220  001001                                  BNE     3$
121222  104000                                  EMT                     ;
121224  042777  000100  177304          3$:     BIC     #BIT6,@TCSR     ;CLEAR BIT6 OF TCSR
121232  032777  000100  177276                  BIT     #BIT6,@TCSR     ;TEST BIT6 OF TCSR
121240  001401                                  BEQ     4$
```

```
121242  104000                          EMT                             ;
121244                          4$:
121244  052777  000100  177264          BIS     #BIT6,@TCSR             ;SET BIT6 OF TCSR
121252  000005                          RESET                           ;CLEAR BIT6 WITH RESET
121254  032777  000100  177254          BIT     #BIT6,@TCSR             ;TEST BIT6 OF TCSR
121262  001401                          BEQ     5$
121264  104000                          EMT                             ;
121266  010377  177254          5$:     MOV     R3,@TVECT               ;RESTORE XMIT VECTOR


                                ;*************************************************************************
                                ;TEST 610       TEST THAT BIT6 OF RCSR CAN BE SET & RESET
                                ;*************************************************************************
121272                          TS610:
121272  017703  177244                  MOV     @RVECT,R3               ;SAVE RECEIVE VECTOR
121276  012777  121320  177236          MOV     #1$,@RVECT              ;SET UP INTERRUPT VECTOR FOR ERROR REPORT
121304  106427  000340                  MTPS    #340                    ;SET PSW TO PRIORITY 7
121310  032777  000100  177214          BIT     #BIT6,@RCSR             ;TEST BIT6 OF RCSR
121316  001401                          BEQ     2$
121320                          1$:
121320  104000                          EMT                             ;
121322  052777  000100  177202  2$:     BIS     #BIT6,@RCSR             ;SET BIT6 OF RCSR
121330  032777  000100  177174          BIT     #BIT6,@RCSR             ;TEST BIT6 OF RCSR
121336  001001                          BNE     3$
121340  104000                          EMT                             ;
121342  042777  000100  177162  3$:     BIC     #BIT6,@RCSR             ;CLEAR BIT6 OF RCSR
121350  032777  000100  177154          BIT     #BIT6,@RCSR             ;TEST BIT6 OF RCSR
121356  001401                          BEQ     4$
121360  104000                          EMT                             ;
121362                          4$:
121362  052777  000100  177142          BIS     #BIT6,@RCSR             ;SET BIT6 OF RCSR
121370  000005                          RESET                           ;CLEAR BIT6OF RCSR WITH RESET
121372  032777  000100  177132          BIT     #BIT6,@RCSR             ;TEST BIT6 OF RCSR
121400  001401                          BEQ     5$
121402  104000                          EMT                             ;
121404  010377  177132          5$:     MOV     R3,@RVECT               ;RESTORE RECEIVE VECTOR



                                ;*************************************************************************
                                ;TEST 611       TEST THAT XMIT INTERRUPTS ONLY WHEN ENABLED
                                ;*************************************************************************
121410                          TS611:
121410  042777  000100  177120          BIC     #BIT6,@TCSR             ;CLEAR TRANSMIT INTERRUPT ENABLE
121416  017703  177124                  MOV     @TVECT,R3               ;SAVE XMIT VECTOR
121422  012777  121444  177116          MOV     #2$,@TVECT              ;POINT XMIT VECTOR TO ERROR REPORT
121430  105777  177102          1$:     TSTB    @TCSR                   ;WAIT FOR DONE
121434  100375                          BPL     1$
121436  106427  000140                  MTPS    #140                    ;SET PSW TO PRIORITY 3
121442  000401                          BR      3$
121444                          2$:
121444  104000                          EMT                             ;
121446  012777  121466  177072  3$:     MOV     #4$,@TVECT              ;SET XMIT VECTOR TO END OF TEST
121454  052777  000100  177054          BIS     #BIT6,@TCSR             ;ENABLE INTERRUPTS
121462  000240                          NOP

```

```
121464  104000                                  EMT                             ;XMIT DID NOT IN    UPT

121466  042777  000100  177042          4$:     BIC     #BIT6,@TCSR             ;DISABLE INTERRUPTS
121474  022626                                  CMP     (SP)+,(SP)+             ;RESTORE SP AFTER INTERRUPT
121476  010377  177044                          MOV     R3,@TVECT               ;RESTORE XMIT VECTOR



                                        ;*******************************************************************************
                                        ;TEST 612       TEST THAT XMIT INTERRUPTS DO NOT OCCUR WHEN DISABLED
                                        ;*******************************************************************************
121502                                  TS612:
121502  042777  000100  177026                  BIC     #BIT6,@TCSR             ;DISABLE INTERRUPTS
121510  106427  000340                          MTPS    #340                    ;SET PSW TO PRIORITY 7
121514  017703  177026                          MOV     @TVECT,R3               ;SAVE XMIT VECTOR
121520  012777  121546  177020                  MOV     #2$,@TVECT              ;POINT XMIT VECTOR TO ERROR REPORT
121526  105777  177004                  1$:     TSTB    @TCSR                   ;WAIT FOR DONE
121532  100375                                  BPL     1$
121534  052777  000100  176774                  BIS     #BIT6,@TCSR             ;ENABLE INTERRUPT
121542  000240                                  NOP
121544  000401                                  BR      3$
121546                                  2$:
121546  104000                                  EMT                             ;
121550  042777  000100  176760          3$:     BIC     #BIT6,@TCSR             ;CLEAR INTERRUPT ENABLE
121556  012777  121574  176762                  MOV     #4$,@TVECT              ;POINT XMIT VECTOR TO ERROR REPORT
121564  106427  000140                          MTPS    #140                    ;SET PSW TO PRIORITY 3
121570  000240                                  NOP
121572  000401                                  BR      5$
121574                                  4$:
121574  104000                                  EMT                             ;
121576  010377  176744                  5$:     MOV     R3,@TVECT               ;RESTORE XMIT VECTOR



                                        ;*******************************************************************************
                                        ;TEST 613       TEST TRANSMITTER FOR DOUBLE INTERRUPTS
                                        ;*******************************************************************************
121602                                  TS613:
121602  042777  000100  176726                  BIC     #BIT6,@TCSR             ;CLEAR INTERRUPT ENABLE
121610  017703  176732                          MOV     @TVECT,R3               ;SAVE XMIT VECTOR
121614  017704  176730                          MOV     @TPSW,R4                ;SAVE XMIT PSW VECTOR
121620  012777  121660  176720                  MOV     #2$,@TVECT              ;SET UP XMIT VECTOR
121626  012777  000340  176714                  MOV     #340,@TPSW              ;SET PIO 7 AFTER INTERRUPT
121634  106427  000140                          MTPS    #140                    ;SET PSW TO PRIORITY 3
121640  105777  176672                  1$:     TSTB    @TCSR                   ;WAIT FOR DONE
121644  100375                                  BPL     1$
121646  052777  000100  176662                  BIS     #BIT6,@TCSR             ;ENABLE INTERRUPTS
121654  000240                                  NOP

121656  104000                                  EMT                             ;
                                                                                ;XMIT INTERRUPT DID NOT OCCUR
121660  022626                          2$:     CMP     (SP)+,(SP)+             ;RESTORE SP AFTER INTERRUPT
121662  012777  121706  176656                  MOV     #4$,@TVECT              ;POINT XMIT VECTOR TO ERROR
121670  106427  000140                          MTPS    #140                    ;SET PSW TO PRIORITY 3
121674  000240                                  NOP                             ;GIVE TIME FOR ANY INTERRUPTS
121676  042777  000100  176632                  BIC     #BIT6,@TCSR             ;DISABLE INTERRUPTS
121704  000401                                  BR      5$
121706                                  4$:
```

```
121706  104000                            EMT                        ;
121710  010377  176632            5$:     MOV     R3,@TVECT          ;RESTORE XMIT VECTOR
121714  010477  176630                    MOV     R4,@TPSW           ;RESTORE XMIT PSW VECTOR
                                  ;*******************************************************************************
                                  ;TEST 614        TEST THAT XMIT INTERRUPT CLEARS WITH LOADING TBUF
                                  ;*******************************************************************************
121720                            TS614:
121720  042777  000100  176610            BIC     #BIT6,@TCSR        ;DISABLE INTERRUPTS
121726  106427  000340                    MTPS    #340               ;SET PSW TO PRIORITY 7
121732  017703  176610                    MOV     @TVECT,R3          ;SAVE XMIT VECTOR
121736  012777  122006  176602            MOV     #2$,@TVECT         ;POINT XMIT VECTOR TO ERROR
121744  052777  000100  176564            BIS     #BIT6,@TCSR        ;ENABLE INTERRUPTS
121752  005077  176562                    CLR     @TBUF              ;LOAD TBUF
121756  105777  176554            1$:     TSTB    @TCSR              ;WAIT FOR DONE (INTERRUPT)
121762  100375                            BPL     1$
121764  005077  176550                    CLR     @TBUF              ;FILL SECOND BUFFER TO RESET INT.
121770  106427  000140                    MTPS    #140               ;SET PSW TO PRIORITY 3
121774  000240                            NOP                        ;GIVE TIME FOR ANY INTERRUPTS
121776  042777  000100  176532            BIC     #BIT6,@TCSR        ;DISABLE INTERRUPTS
122004  000401                            BR      3$
122006                            2$:
122006  104000                            EMT                        ;
122010  010377  176532            3$:     MOV     R3,@TVECT          ;RESTORE XMIT VECTOR
122014  005000                            CLR     R0                 ;INITIALIZE LOOP COUNTER
122016  005200                    4$:     INC     R0                 ;INCREMENT LOOP COUNTER
122020  001376                            BNE     4$                 ; UNTIL COUNTER = 0
122022  005777  176506                    TST     @RBUF              ;CLEAR RECEIVER BUFFER
                                  ;*******************************************************************************
                                  ;TEST 615        TEST THAT RCVR DONE (7) SET & CLEAR PROPERLY
                                  ;*******************************************************************************
122026                            TS615:
122026  052777  000004  176502            BIS     #BIT2,@TCSR        ;SET MAINTENANCE WRAP
122034  005000                            CLR     R0                 ;CLEAR A TIMER
122036  005077  176476                    CLR     @TBUF              ;LOAD TRANSMIT BUFFER
122042  105777  176464            WDONE1: TSTB    @RCSR              ;CHECK FOR RECEIVER DONE
122046  100406                            BMI     6$                 ;BR, IF DONE
122050  005200                            INC     R0                 ;INCREMENT TIMER, IF NOT DONE
122052  001373                            BNE     WDONE1             ;CONTINUE WAIT IF TIME REMAINS
122054  042777  000004  176454            BIC     #BIT2,@TCSR        ;CLEAR MAINTENANCE BIT
122062  104000                            EMT                        ;RECEIVER DONE NEVER SET
122064  000005                    6$:     RESET                      ;CLEAR DONE, MAINTENANCE MODE AND RECEIVER
                                                                     ;   BUFFER WITH RESET
122066  105777  176440                    TSTB    @RCSR              ;CHECK FOR DONE CLEAR
122072  001401                            BEQ     7$
122074  104000                            EMT                        ;RESET DID NOT CLEAR RCVR DONE
122076                            7$:
                                  ;*******************************************************************************
                                  ;TEST 616        TEST THAT READING RBUF CLEARS RECEIVER DONE
                                  ;*******************************************************************************
```

```
122076                              TS616:
122076  052777  000004  176432             BIS     #BIT2,@TCSR      ;SET MAINTENANCE WRAP
122104  005077  176430                     CLR     @TBUF            ;LOAD TRANSMITTER
122110  105777  176416              1$:    TSTB    @RCSR            ;WAIT FOR RECEIVER DONE
122114  100375                             BPL     1$
122116  017700  176412                     MOV     @RBUF,R0         ;READ RECEIVE BUFFER
122122  042777  000004  176406             BIC     #BIT2,@TCSR      ;CLEAR MAINTENANCE BIT
122130  105777  176376                     TSTB    @RCSR            ;CHECK FOR RECEIVE DONE CLEAR
122134  001401                             BEQ     TS617
122136  104000                             EMT                      ;
                                                                    ;READING RBUF DID NOT CLEAR RCVR DONE


                                    ;**********************************************************************
                                    ;TEST 617      TEST THAT RCVR INTERRUPTS ONLY WHEN ENABLED
                                    ;**********************************************************************
122140                              TS617:
122140  042777  000100  176370             BIC     #BIT6,@TCSR      ;DISABLE TRANSMIT INTERRUPTS
122146  042777  000100  176356             BIC     #BIT6,@RCSR      ;DISABLE RECEIVER INTERRUPTS
122154  052777  000004  176354             BIS     #BIT2,@TCSR      ;SET MAINTENANCE WRAP
122162  017703  176354                     MOV     @RVECT,R3        ;SAVE RECEIVE VECTOR
122166  012777  122222  176346             MOV     #2$,@RVECT       ;POINT RCV VECTOR TO ERROR REPORT
122174  106427  000140                     MTPS    #140             ;SET PSW TO PRIORITY 3
122200  005077  176334                     CLR     @TBUF            ;SEND A CHARACTER
122204  105777  176322              1$:    TSTB    @RCSR            ;WAIT FOR RECEIVER DONE
122210  100375                             BPL     1$
122212  042777  000004  176316             BIC     #BIT2,@TCSR      ;CLEAR MAINTENANCE BIT
122220  000404                             BR      3$               ;CONTINUE TEST

122222  042777  000004  176306      2$:    BIC     #BIT2,@TCSR      ;CLEAR MAINTENANCE BIT
122230  104000                             EMT                      ;
122232  012777  122252  176302      3$:    MOV     #4$,@RVECT       ;POINT RCV VECTOR TO END OF TEST
122240  052777  000100  176264             BIS     #BIT6,@RCSR      ;ENABLE RCV INTERRUPTS
122246  000240                             NOP                      ;GIVE ANY INTERRUPTS TIME
122250  104000                             EMT                      ;
122252  042777  000100  176252      4$:    BIC     #BIT6,@RCSR      ;DISABLE INTERRUPTS
122260  022626                             CMP     (SP)+,(SP)+      ;RESTORE SP AFTER INTERRUPT
122262  010377  176254                     MOV     R3,@RVECT        ;RESTORE RECEIVE VECTOR
122266  005777  176242                     TST     @RBUF            ;CLEAR RECEIVER BUFFER


                                    ;**********************************************************************
                                    ;TEST 620      TEST THAT RCVR INTERRUPTS DO NOT OCCUR WHEN DISABLED
                                    ;**********************************************************************
122272                              TS620:
122272  106427  000340                     MTPS    #340             ;SET PSW TO PRIORITY 7
122276  017703  176240                     MOV     @RVECT,R3        ;SAVE RECEIVE VECTOR
122302  012777  122342  176232             MOV     #2$,@RVECT       ;POINT RCVR VECTOR TO ERROR REPORT
122310  052777  000004  176220             BIS     #BIT2,@TCSR      ;SET MAINTENANCE WRAP
122316  005077  176216                     CLR     @TBUF            ;SEND A CHARACTER
122322  105777  176204              1$:    TSTB    @RCSR            ;WAIT FOR RECEIVER DONE
122326  100375                             BPL     1$
122330  052777  000100  176174             BIS     #BIT6,@RCSR      ;ENABLE INTERRUPTS
122336  000240                             NOP                      ;GIVE TIME FOR INTERRUPT
122340  000404                             BR      3$               ;CONTINUE TEST
122342  042777  000004  176166      2$:    BIC     #BIT2,@TCSR      ;CLEAR MAINTENANCE BIT
```

```
122350  104000                          EMT                             ;
122352  042777  000100  176152  3$:     BIC     #BIT6,@RCSR             ;CLEAR INTERRUPT ENABLE
122360  012777  122404  176154          MOV     #4$,@RVECT              ;POINT RCVR VECTOR TO ERROR REPORT
122366  106427  000140                  MTPS    #140                    ;SET PSW TO PRIORITY 3
122372  000240                          NOP                             ;GIVE TIME FOR ANY INTERRUPT
122374  042777  000004  176134          BIC     #BIT2,@TCSR             ;CLEAR MAINTENANCE BIT
122402  000404                          BR      5$                      ;BR TO END OF TEST, IF NO INTERRUPT

122404  042777  000004  176124  4$:     BIC     #BIT2,@TCSR             ;CLEAR MAINTENANCE BIT
122412  104000                          EMT                             ;
122414  010377  176122          5$:     MOV     R3,@RVECT               ;RESTORE RECEIVE VECTOR
122420  005777  176110                  TST     @RBUF                   ;CLEAR RECEIVER BUFFER


                                ;*******************************************************************************
                                ;TEST 621       TEST RECEIVER FOR DOUBLE INTERRUPTS
                                ;*******************************************************************************
122424                          TS621:
122424  017703  176112                  MOV     @RVECT,R3               ;SAVE RFCEIVE VECTOR
122430  017704  176110                  MOV     @RPSW,R4                ;SAVE RECEIVE PSW VECTOR
122434  012777  122514  176100          MOV     #2$,@RVECT              ;POINT RCV VECTOR TO CONTINUE TEST
122442  012777  000340  176074          MOV     #340,@RPSW              ;SET PRIORITY TO 7 AFTER INTERRUPT
122450  106427  000140                  MTPS    #140                    ;SET PSW TO PRIORITY 3
122454  052777  000004  176054          BIS     #BIT2,@TCSR             ;SET MAINTENANCE WRAP
122462  005077  176052                  CLR     @TBUF                   ;SEND A CHARACTER
122466  105777  176040          1$:     TSTB    @RCSR                   ;WAIT FOR RCVR DONE
122472  100375                          BPL     1$
122474  042777  000004  176034          BIC     #BIT2,@TCSR             ;CLEAR MAINTENANCE BIT
122502  052777  000100  176022          BIS     #BIT6,@RCSR             ;ENABLE RCV INTERRUPTS
122510  000240                          NOP                             ;GIVE SOME TIME
122512  104000                          EMT                             ;
122514  022626                  2$:     CMP     (SP)+,(SP)+             ;RESTORE SP AFTER INTERRUPT
122516  012777  122552  176016          MOV     #3$,@RVECT              ;POINT RCV VECTOR TO ERROR REPORT
122524  106427  000140                  MTPS    #140                    ;SET PSW TO PRIORITY 7
122530  000240                          NOP                             ;GIVE SOME TIME
122532  042777  000100  175772          BIC     #BIT6,@RCSR             ;CLEAR INTERRUPT ENABLE
122540  010377  175776                  MOV     R3,@RVECT               ;RESTORE RECEIVE VECTOR
122544  010477  175774                  MOV     R4,@RPSW                ;RESTORE RECEIVE PSW VECTOR
122550  000401                          BR      4$
122552                          3$:
122552  104000                          EMT                             ;
122554  010377  175762          4$:     MOV     R3,@RVECT               ;RESTORE RECEIVE VECTOR
122560  005777  175750                  TST     @RBUF                   ;CLEAR RECEIVER BUFFER


                                ;*******************************************************************************
                                ;TEST 622       TFST THAT RCVR INTERRUPT CLEARS BY READING RBUF
                                ;*******************************************************************************
122564                          TS622:
122564  106427  000340                  MTPS    #340                    ;SET PSW TO PRIORITY 7
122570  017703  175746                  MOV     @RVECT,R3               ;SAVE RECEIVE VECTOR
122574  012777  122660  175740          MOV     #2$,@RVECT              ;POINT RCV VECTOR TO ERROR REPORT
122602  052777  000100  175722          BIS     #BIT6,@RCSR             ;SET RCVR INTERRUPT ENABLE
122610  052777  000004  175720          BIS     #BIT2,@TCSR             ;SET MAINTENANCE WRAP
122616  005077  175716                  CLR     @TBUF                   ;SEND A CHARACTER
122622  105777  175704          1$:     TSTB    @RCSR                   ;WAIT FOR DONE (INTERRUPT)
```

```
 122626  100375                                  BPL     1$
 122630  042777  000004  175700                  BIC     #BIT2,@TCSR       ;CLEAR MAINTENANCE BIT
 122636  005777  175672                          TST     @RBUF             ;READ RBUF TO CLEAR PENDING INTERRPUT
 122642  106427  000140                          MTPS    #140              ;SET PSW TO PRIORITY 3
 122646  000240                                  NOP                       ;ALLOW TIME FOR ANY ERRONEOUS INTERRUPT
 122650  042777  000100  175654                  BIC     #BIT6,@RCSR       ;NO INTERRUPT-CLEAR INT. ENABLE
 122656  000401                                  BR      3$
 122660                                  2$:
 122660  104000                                  EMT                       ;
 122662  010377  175654                  3$:     MOV     R3,@RVECT         ;RESTORE RECEIVE VECTOR



                                         ;*************************************************************************
                                         ;TEST 623       TEST THAT RESET CLEARS RECEIVE INTERRUPT
                                         ;*************************************************************************
 122666                                  TS623:
 122666  106427  000340                          MTPS    #340              ;SET PSW TO PRIORITY 7
 122672  017703  175644                          MOV     @RVECT,R3         ;SAVE RECEIVE VECTOR
 122676  012777  122762  175636                  MOV     #2$,@RVECT        ;POINT RCV VECTOR TO ERROR REPORT
 122704  052777  000100  175620                  BIS     #BIT6,@RCSR       ;SET RCV INTERRUPT ENABLE
 122712  052777  000004  175616                  BIS     #BIT2,@TCSR       ;SET MAINTENANCE WRAP
 122720  012777  000377  175612                  MOV     #377,@TBUF        ;SEND AN ALL 1'S CHARACTER
 122726  105777  175600                  1$:     TSTB    @RCSR             ;WAIT FOR RCV DONE
 122732  100375                                  BPL     1$
 122734  000005                                  RESET                     ;CLEAR RCV INTERRUPT & RBUF
 122736  052777  000100  175566                  BIS     #BIT6,@RCSR       ;TURN ON RECEIVER INTERRUPT
 122744  106427  000140                          MTPS    #140              ;SET PSW TO PRIORITY 3
 122750  000240                                  NOP                       ;ALLOW TIME FOR AN ERRONEOUS INTERRUPT
 122752  042777  000100  175552                  BIC     #BIT6,@RCSR       ;NO INTERRUPT-CLEAR INT. ENABLE
 122760  000401                                  BR      3$
 122762                                  2$:
 122762  104000                                  EMT                       ;
 122764  010377  175552                  3$:     MOV     R3,@RVECT         ;RESTORE RECEIVE VECTOR


                                         ;*************************************************************************
                                         ;TEST 624       TEST THAT THE "OR" ERROR (BIT14) & "ERROR" (BIT15) CAN BE SET
                                         ;*************************************************************************
 122770                                  TS624:
 122770  052777  000004  175540                  BIS     #BIT2,@TCSR       ;SET MAINTENANCE WRAP
 122776  012700  000003                          MOV     #3,R0             ;SET CHARACTER COUNT TO SEND 3 CHAR.
 123002  005077  175532                  1$:     CLR     @TBUF             ;LOAD TRANSMIT BUFFER
 123006  105777  175524                  2$:     TSTB    @TCSR             ;WAIT FOR TRANSMIT DONE
 123012  100375                                  BPL     2$
 123014  005300                                  DEC     R0                ;DECREMENT CHARACTER COUNT
 123016  001371                                  BNE     1$                ;BR IF ALL CHARACTERS NOT TRANSMITTED
 123020  042777  000004  175510                  BIC     #BIT2,@TCSR       ;CLEAR MAINTENANCE BIT
 123026  032777  040000  175500                  BIT     #BIT14,@RBUF      ;TEST FOR "OR" ERROR FLAG
 123034  001001                                  BNE     3$
 123036  104000                                  EMT                       ;
 123040  032777  100000  175466  3$:     BIT     #BIT15,@RBUF      ;TEST "ERROR" FLAG
 123046  001001                                  BNE     4$
 123050  104000                                  EMT                       ;
 123052  005000                          4$:     CLR     R0                ;INITIALIZE LOOP COUNTER
 123054  005200                          5$:     INC     R0                ;INCREMENT LOOP COUNTER
```

```
123056  001376                          BNE     5$              ;  UNTIL COUNTER = 0
123060  005777  175450                  TST     @RBUF           ;CLEAR RECEIVER BUFFER


                                ;*******************************************************************************
                                ;TEST 625       TEST DATA PATH FROM TRANSMITTER TO RECEIVER USING MAINTENANCE WRAP
                                ;*******************************************************************************
123064                          TS625:
123064  000005                          RESET                   ;CLEAR THE WORLD
123066  005001                          CLR     R1              ;CLEAR REGISTER FOR TEST DATA
123070  052777  000004  175440          BIS     #BIT2,@TCSR     ;SET MAINTENANCE WRAP
123076  105201                  1$:     INCB    R1              ;INCREMENT THE TEST DATA
123100  110177  175434                  MOVB    R1,@TBUF        ;XMIT A CHARACTER
123104  105777  175422          2$:     TSTB    @RCSR           ;WAIT FOR RECEIVER DONE
123110  100375                          BPL     2$
123112  017702  175416                  MOV     @RBUF,R2        ;GET RECEIVED CHARACTER
123116  020102                          CMP     R1,R2           ;COMPARE DATA
123120  001404                          BEQ     3$              ;BR, IF NON-COMPARE
123122  042777  000004  175406          BIC     #BIT2,@TCSR     ;CLEAR MAINTENANCE BIT
123130  104000                          EMT                     ;
123132  105701                  3$:     TSTB    R1              ;TEST XMIT DATA FOR ZERO
123134  001360                          BNE     1$              ;IF ALL DATA HES NOT BEEN CHECKED GO BACK
123136  042777  000004  175372          BIC     #BIT2,@TCSR     ;CLEAR MAINTENANCE BIT
123144  000167  000050                  JMP     KWSTRT          ;GET TO NEXT TEST

123150  012737  123202  000004  ERROR5: MOV     #1$,@#4         ;SET UP THE TIMEOUT VECTOR
123156  012737  000340  000006          MOV     #340,@#6
123164  005737  177766                  TST     @#177766        ;CHECK TO SEE IF THE POWER FAILED,WILL TRAP
                                                                ;TO 1$ IF UNIBUS MAP BOARD IS NOT PRESENT
                                                                ;SINCE THE CPUERR(177766)IS ON THE UNIBUS
                                                                ;MAP BOARD
123170  001404                          BEQ     1$              ;NO,GO REPORT THE ERROR
123172  012737  000177  001002          MOV     #177,@#$FATAL   ;YES,FLAG THAT THE POWER FAILED
123200  000403                          BR      2$              ; AND GO SET FATAL ERROR FLAG
123202  012737  000004  001002  1$:     MOV     #4,@#$FATAL     ;SET UP FATAL ERROR NUMBER
123210  012767  000001  055562  2$:     MOV     #1,$MSGTY       ;SET FATAL ERROR FLAG
123216  000777                  SL1HLT: BR      .


123220  032777  000010  076076  KWSTRT: BIT     #10,@SWR
123226  001402                          BEQ     1$
123230  000167  001246                  JMP     SLU2ST
123234  012737  000005  001004  1$:     MOV     #5,@#$TESTN     ;PUT TEST NUMBER IN MAILBOX
123242  012737  124412  000030          MOV     #ERROR6,@#30    ;SET UP ERROR CALL

123250                          LKSTST:
                                ;*******************************************************************************
                                ;TEST 626       TEST ABILITY TO ACCESS LKS
                                ;*******************************************************************************
123250                          TS626:
123250  013703  000004                  MOV     @#4,R3          ;SAVE TIMEOUT VECTOR
123254  012737  123270  000004          MOV     #1$,@#4         ;SET UP TIMEOUT VECTOR
123262  005777  175264                  TST     @LKS            ;ACCESS LKS
123266  000401                          BR      2$
123270                          1$:
123270  104000                          EMT                     ;
```

```
123272 010337 000004            2$:     MOV     R3,@#4          ;RESTORE TIMEOUT VECTOR

                                ;*******************************************************************************
                                ;TEST 627       TEST THAT BIT6 OF LKS CAN BE SET & RESET
                                ;*******************************************************************************
123276                          TS627:
123276 017703 175252                    MOV     @RTCVT,R3       ;SAVE LINE CLOCK VECTOR
123302 012777 123324 175244             MOV     #1$,@RTCVT      ;SET UP INTERRUPT VECTOR FOR ERROR REPORT
123310 106427 000340                    MTPS    #340            ;SET PSW TO PRIORITY 7
123314 032777 000100 175230             BIT     #BIT6,@LKS      ;TEST BIT6 OF LKS
123322 001401                           BEQ     2$
123324                          1$:
123324 104000                           EMT                     ;
123326 052777 000100 175216     2$:     BIS     #BIT6,@LKS      ;SET BIT6 OF LKS
123334 032777 000100 175210             BIT     #BIT6,@LKS      ;TEST BIT6 OF LKS
123342 001001                           BNE     3$
123344 104000                           EMT                     ;
123346 042777 000100 175176     3$:     BIC     #BIT6,@LKS      ;CLEAR BIT6 OF LKS
123354 032777 000100 175170             BIT     #BIT6,@LKS      ;TEST BIT6 OF LK
123362 001401                           BEQ     4$
123364 104000                           EMT                     ;
123366 032737 000001 001020     4$:     BIT     #1,@#$ENV       ;ARE WE RUNNING UNDER APT
123374 001403                           BEQ     70$             ;IF NO THEN DO TEST
123376 005737 001006                    TST     @#$PASS         ;IS THIS FIRST PASS
123402 001011                           BNE     5$              ;IF NO SKIP TO TEST END
123404                          70$:
123404 052777 000100 175140             BIS     #BIT6,@LKS      ;SET BIT6 OF LKS
123412 000005                           RESET                   ;CLEAR BIT6 OF LKS WITH PESET
123414 032777 000100 175130             BIT     #BIT6,@LKS      ;TEST BIT6 OF LKS
123422 001401                           BEQ     5$
123424 104000                           EMT                     ;
123426 010377 175122            5$:     MOV     R3,@RTCVT       ;RESTORE LINE CLOCK VECTOR

                                ;*******************************************************************************
                                ;TEST 630       TEST THAT BIT7 OF LKS SETS & CAN BE CLEARED
                                ;*******************************************************************************
123432                          TS630:
123432                          1$:
123432 032737 000001 001020             BIT     #1, @#$ENV      ;ARE WE RUNNING UNDER APT
123440 001403                           BEQ     70$             ;IF NO THEN DO TEST
123442 005737 001006                    TST     @#$PASS         ;IS THIS FIRST PASS
123446 001032                           BNE     TS631           ;IF NO THEN SHIP TO NEXT TEST
123450                          70$:
123450 105777 175076                    TSTB    @LKS            ;TEST FOR BIT7 OF LKS
123454 100401                           BMI     2$
123456 104000                           EMT                     ;
123460 042777 000200 175064     2$:     BIC     #BIT7,@LKS      ;CLEAR BIT7 OF LKS
123466 032777 000200 175056             BIT     #BIT7,@LKS      ;TEST BIT7 OF LKS
123474 001410                           BEQ     3$
123476 042777 000200 175046             BIC     #BIT7,@LKS      ;TRY ONE MOR TIME BECAUSE THE CLOCK MAY
123504 032777 000200 175040             BIT     #BIT7,@LKS      ;MAY HAVE SET IMMEDIATELY AFTER THE FIRST CLEAR
123512 001401                           BEQ     3$
123514 104000                           EMT                     ;
123516 005000                   3$:     CLR     R0              ;CLEAR TIMER
123520 105777 175026            CONT1:  TSTB    @LKS            ;TEST FOR BIT7 OF LKS
123524 100403                           BMI     1$              ;BR. IF SET
```

```
123526 005200                                   INC     R0              ;INCREMENT TIMER
123530 001373                                   BNE     CONT1
123532 104000                                   EMT                     ;
123534                                  1$:


                                        ;**********************************************************************
                                        ;TEST 631       TEST THAT THE REAL TIME CLOCK INTERRUPTS PROPERLY
                                        ;**********************************************************************
123534                                  TS631:
123534 106427 000340                            MTPS    #340            ;SET PSW TO PRIORITY 7
123540 017703 175010                            MOV     @RTCVT,R3       ;SAVE LINE CLOCK VECTOR
123544 017704 175006                            MOV     @RTCPSW,R4      ;SAVE LINE CLOCK PSW VECTOR
123550 012777 123612 174776                     MOV     #2$,@RTCVT      ;SET RTC INTERRUPT VECTOR TO ERROR REPORT
123556 012777 000340 174772                     MOV     #340,@RTCPSW    ;KEEP PRIORITY AT 7
123564 042777 000200 174760                     BIC     #BIT7,@LKS      ;CLEAR CLOCK DONE FLAG
123572 052777 000100 174752                     BIS     #BIT6,@LKS      ;SET INTERRUPT ENABLE
123600 105777 174746                    1$:     TSTB    @LKS            ;WAIT FOR RTC DONE(INTERRUPT REQUEST)
123604 100375                                   BPL     1$
123606 000240                                   NOP                     ;GIVE TIME FOR ANY INTERRUPTS
123610 000401                                   BR      3$
123612                                  2$:
123612 104000                                   EMT                     ;
123614 005077 174732                    3$:     CLR     @LKS            ;DISABLE RTC INTERRUPTS & CLEAR DONE
123620 012777 123644 174726                     MOV     #4$,@RTCVT      ;SET RTC INTERRUPT VECTOR FOR ERROR
123626 106427 000240                            MTPS    #240            ;CHANGE PSW TO PRIORITY 5
123632 105777 174714                    20$:    TSTB    @LKS            ;WAIT FOR DONE (INTERRUPT REQUEST)
123636 100375                                   BPL     20$
123640 000240                                   NOP                     ;GIVE TIME FOR ANY INTERRUPT
123642 000401                                   BR      5$
123644                                  4$:
123644 104000                                   EMT                     ;
123646 012777 123702 174700             5$:     MOV     #7$,@RTCVT      ;POINT RTC VECTOR TO END OF TEST
123654 042777 000200 174670                     BIC     #BIT7,@LKS      ;CLEAR CLOCK DONE FLAG
123662 052777 000100 174662                     BIS     #BIT6,@LKS      ;ALLOW INTERRUPTS
123670 105777 174656                    6$:     TSTB    @LKS            ;WAIT FOR RTC DONE
123674 100375                                   BPL     6$
123676 000240                                   NOP                     ;GIVE TIME FOR INTERRUPT

123700 104000                                   EMT                     ;RTC INTERRUPT DID NOT OCCUR

123702 022626                           7$:     CMP     (SP)+,(SP)+     ;RESTORE SP AFTER INTERRUPT
123704 042777 000100 174640                     BIC     #BIT6,@LKS      ;DISABLE INTERRUPTS
123712 010377 174636                            MOV     R3,@RTCVT       ;RESTORE LINE CLOCK VECTOR
123716 010477 174634                            MOV     R4,@RTCPSW      ;RESTORE LINE CLOCK PSW VECTOR





                                        ;**********************************************************************
                                        ;TEST 632       TEST RTC FOR DOUBLE INTERRUPTS
                                        ;**********************************************************************
123722                                  TS632:
123722 032737 000001 001020                     BIT     #1, @#$ENV      ;ARE WE RUNNING UNDER APT
123730 001403                                   BEQ     70$             ;IF NO THEN DO TEST
```

```
123732  005737  001006                  TST     @#$PASS         ;IS THIS FIRST PASS
123736  001047                          BNE     TS633           ;IF NO THEN SHIP TO NEXT TFST
123740                          70$:
123740  017703  174610                  MOV     @RTCVT,R3       ;SAVE LINE CLOCK VECTOR
123744  017704  174606                  MOV     @RTCPSW,R4      ;SAVE LINE CLOCK PSW VECTOR
123750  012777  124016  174576          MOV     #2$,@RTCVT      ;SET UP RTC INTERRUPT VECTOR
123756  012777  000340  174572          MOV     #340,@RTCPSW    ;DISALLOW INTERRUPTS AFTER THE INTERRUPT
123764  106427  000240                  MTPS    #240            ;SET PSW TO PRIORITY 5
123770  042777  000200  174554          BIC     #BIT7,@LKS      ;CLEAR CLOCK DONE FLAG
123776  052777  000100  174546          BIS     #BIT6,@LKS      ;ENABLE CLOCK INTERRUPTS
124004  105777  174542          1$:     TSTB    @LKS            ;WAIT FOR DONE
124010  100375                          BPL     1$
124012  000240                          NOP                     ;GIVE TIME FOR ANY INTERRUPT

124014  104000                          EMT                     ;RTC INTERRUPT DID NOT OCCUR

124016  022626                  2$:     CMP     (SP)+,(SP)+     ;RESTORE SP AFTER INTERRUPT
124020  012777  124036  174526          MOV     #3$,@RTCVT      ;POINT RTC VECTOR TO ERROR REPORT
124026  106427  000240                  MTPS    #240            ;SET PSW TO PRIORITY 5
124032  000240                          NOP                     ;GIVE SOME TIME FOR AN INTERRUPT
124034  000401                          BR      4$
124036                          3$:
124036  104000                          EMT                     ;
124040  042777  000100  174504  4$:     BIC     #BIT6,@LKS      ;DISABLE CLOCK INTERRUPTS
124046  010377  174502                  MOV     R3,@RTCVT       ;RESTORE LINE CLOCK VECTOR
124052  010477  174500                  MOV     R4,@RTCPSW      ;RESTORE LINE CLOCK PSW VECTOR


                                ;*******************************************************************************
                                ;TEST 633       TEST THAT RTC INTERRUPT CLEARS WITH RESET
                                ;*******************************************************************************
124056                          TS633:
124056  032737  000001  001020          BIT     #1, @#$ENV      ;ARE WE RUNNING UNDER APT
124064  001403                          BEQ     70$             ;IF NO THEN DO TEST
124066  005737  001006                  TST     @#$PASS         ;IS THIS FIRST PASS
124072  001033                          BNE     TS634           ;IF NO THEN SHIP TO NEXT TEST
124074                          70$:
124074  106427  000340                  MTPS    #340            ;SET PSW TO PRIORITY 7
124100  017703  174450                  MOV     @RTCVT,R3       ;SAVE LINE CLOCK VECTOR
124104  012777  124154  174442          MOV     #2$,@RTCVT      ;POINT RTC VECTOR TO ERROR REPORT
124112  042777  000200  174432          BIC     #BIT7,@LKS      ;CLEAR CLOCK DONE FLAG
124120  052777  000100  174424          BIS     #BIT6,@LKS      ;ENABLE CLOCK INTERRUPTS
124126  105777  174420          1$:     TSTB    @LKS            ;WAIT FOR DONE (INTERRUPT REQUEST)
124132  100375                          BPL     1$
124134  000005                          RESET                   ;CLEAR PENDING INTERRUPT WITH RESET
124136  106427  000240                  MTPS    #240            ;SET PSW TO PRIORITY 5
124142  000240                          NOP                     ;GIVE TIME FOR ANY INTERRUPT
124144  042777  000100  174400          BIC     #BIT6,@LKS      ;DISALLOW INTERRUPTS
124152  000401                          BR      3$
124154                          2$:
124154  104000                          EMT                     ;
124156  010377  174372          3$:     MOV     R3,@RTCVT       ;RESTORE LINE CLOCK VECTOR


                                ;*******************************************************************************
                                ;TEST 634       TEST THAT RTC INTERRUPT CLEARS BY CLEARING BIT7 OF LKS
```

```
                                        ;******************************************************************* ********
124162                                  TS634:
124162  106427  000340                          MTPS    #340            ;SET PSW TO PRIORITY 7
124166  017703  174362                          MOV     @RTCVT,R3       ;SAVE LINE CLOCK VECTOR
124172  012777  124246  174354                  MOV     #2$,@RTCVT      ;POINT RTC VECTOR TO ERROR REPORT
124200  042777  000200  174344                  BIC     #BIT7,@LKS      ;CLEAR CLOCK DONE FLAG
124206  052777  000100  174336                  BIS     #BIT6,@LKS      ;ENABLE CLOCK INTERRUPTS
124214  105777  174332                  1$:     TSTB    @LKS            ;WAIT FOR DONE (INTERRUPT REQUEST)
124220  100375                                  BPL     1$
124222  042777  000200  174322                  BIC     #BIT7,@LKS      ;CLEAR DONE & INTERRUPT
124230  106427  000240                          MTPS    #240            ;SET PSW TO PRIORITY 5
124234  000240                                  NOP                     ;GIVE TIME FOR ANY INTERRUPT
124236  042777  000100  174306                  BIC     #BIT6,@LKS      ;DISALLOW INTERRUPTS
124244  000401                                  BR      3$
124246                                  2$:
124246  104000                                  EMT                     ;

124250  010377  174300                  3$:     MOV     R3,@RTCVT       ;RESTORE LINE CLOCK VECTOR
124254  106427  000340                          MTPS    #340            ;SET PSW TO PRIORITY 7


                                        ;*****************************************************************************
                                        ;TEST 635       TEST CLOCK REPEATABILITY
                                        ;*****************************************************************************
124260                                  TS635:
124260  032737  000001  001020                  BIT     #1,@#$ENV       ;ARE WE RUNNING UNDER APT
124266  001403                                  BEQ     70$             ;IF NO GO DO TEST
124270  005737  001006                          TST     @#$PASS         ;IS THIS FIRST PASS
124274  001102                                  BNE     SLU2ST          ;IF NO THEN SKIP REST OF RTC TESTING
124276                                  70$:
124276  042777  000100  174246                  BIC     #BIT6,@LKS      ;DISALLOW INTERRUPTS

124304  005000                                  CLR     R0              ;CLEAR A TIMER
124306  012701  177777                          MOV     #-1,R1          ;SET A FLAG INDICATING FIRST PASS THRU THIS LOOP
124312  005002                          1$:     CLR     R2              ;CLEAR CLOCK COUNTER
124314  005077  174232                          CLR     @LKS            ;CLEAR DONE
124320  105777  174226                  2$:     TSTB    @LKS            ;SYNC ON DONE
124324  100375                                  BPL     2$
124326  005077  174220                          CLR     @LKS            ;CLEAR DONE
124332  105777  174214                  3$:     TSTB    @LKS            ;IS CLOCK DONE?
124336  100003                                  BPL     4$              ;BR IF NOT , TO INCREMENT TIMER
124340  005202                                  INC     R2              ;IF DONE, INCREMENT CLOCK COUNT
124342  005077  174204                          CLR     @LKS            ;CLEAR DONE
124346  005200                          4$:     INC     R0              ;INCREMENT TIMER
124350  001370                                  BNE     3$              ;BR IF TIME REMAINS
124352  005201                                  INC     R1              ;INCREMENT LOOP PASS FLAG
124354  001003                                  BNE     CMPARE          ;BR IF TWO PASSES HAVE BEEN MADE
124356  010267  000024                          MOV     R2,KFIRST       ;IF NOT, STORE FIRST CLOCK COUNT
124362  000753                                  BR      1$              ;DO LOOP AGAIN
124364  016701  000016                  CMPARE: MOV     KFIRST,R1       ;RECALL FIRST CLOCK COUNT
124370  160201                                  SUB     R2,R1           ;CALCULATE DIFFERENCE OF TWO COUNTS
124372  100001                                  BPL     TOLER           ;IF POSITIVE,SKIP NEGATION OF DIFFERENCE
124374  005401                                  NEG     R1              ;MAKE DIFFERENCE A POSITIVE NUMBER
124376  020127  000002                  TOLER:  CMP     R1,#2           ;COMPARE DIFFERENCE WITH DESIRED TOLERANCE
124402  003437                                  BLE     SLU2ST
124404  104000                                  EMT                     ;
```

```

124406  000000                          KFIRST: .WORD   0
124410  000000                          SECND:  .WORD   0


124412  012737  124444  000004          ERROR6: MOV     #1$,@#4         ;SET UP THE TIMEOUT VECTOR
124420  012737  000340  000006                  MOV     #340,@#6
124426  005737  177766                          TST     @#177766        ;CHECK TO SEE IF THE POWER FAILED,WILL TRAP
                                                                        ;TO 1$ IF UNIBUS MAP BOARD IS NOT PRESENT
                                                                        ;SINCE THE CPUERR(177766)IS ON THE UNIBUS
                                                                        ;MAP BOARD
124432  001404                                  BEQ     1$              ;NO,GO REPORT THE ERROR
124434  012737  000177  001002                  MOV     #177,@#$FATAL   ;YES,FLAG THAT THE POWER FAILED
124442  000403                                  BR      2$              ; AND GO SET FATAL ERROR FLAG
124444  012737  000005  001002          1$:     MOV     #5,@#$FATAL                     ;SET UP FATAL ERROR NUMBER
124452  012767  000001  054320          2$:     MOV     #1,$MSGTY                       ;SET FATAL ERROR FLAG
124460  000777                          LTCHLT: BR      .

                                        ;SERIAL LINE UNIT REGISTER AND VECTOR ADDRESSES FOR SLU2

124462  176500                          RCSR2:  176500                  ;ADDRESS OF RECEIVER COMMAND/STATUS REGISTER
124464  176502                          RBUF2:  176502                  ;ADDRESS OF RECEIVER BUFFER
124466  176504                          TCSR2:  176504                  ;ADDRESS OF TRANSMITTER COMMAND/STATUS REGISTER
124470  176506                          TBUF2:  176506                  ;ADDRESS OF TRANSMITTER BUFFER
124472  000300                          RVECT2: 300                     ;RECEIVER INTERPUPT VECTOR
124474  000302                          RPSW2:  302
124476  000304                          TVECT2: 304                     ;TRANSMITTER INTERRUPT VECTOR
124500  000306                          TPSW2:  306

124502  032777  000020  074614          SLU2ST: BIT     #20,@SWR
124510  001402                                  BEQ     1$
124512  000167  002610                          JMP     UNIQUE
124516  012737  000006  001004          1$:     MOV     #6,@#$TESTN     ;PUT TEST NUMBER IN MAILBOX
124524  012737  127234  000030                  MOV     #ERROR7,@#30    ;SET UP FOR CORRECT ERROR CALL


                                        ;*******************************************************************
                                        ;TEST 636       TEST ABILITY TO REFERENCE TCSR2
                                        ;*******************************************************************
124532                                  TS636:
124532  013703  000004                          MOV     @#4,R3          ;SAVE TIMEOUT VECTOR
124536  012737  124552  000004                  MOV     #1$,@#4         ;SET UP TIMEOUT VECTOR
124544  005777  177716                          TST     @TCSR2          ;REFERENCE THE XMIT COMMAND/STATUS REG.
124550  000401                                  BR      4$
124552                                  1$:
124552  104000                                  EMT                     ;
124554  010337  000004                  4$:     MOV     R3,@#4          ;RESTORE TIMEOUT VECTOR



                                        ;*******************************************************************
                                        ;TEST 637       TEST ABILITY TO REFERENCE TBUF2
                                        ;*******************************************************************
124560                                  TS637:
124560  013703  000004                          MOV     @#4,R3          ;SAVE TIMEOUT VECTOR
124564  012737  124600  000004                  MOV     #1$,@#4         ;SET UP TIMEOUT VECTOR
```

```
124572  005777  177672                  TST     @TBUF2          ;REFERENCE THE XMIT BUFFER
124576  000401                          BR      4$
124600                          1$:
124600  104000                          EMT                     ;
124602  010337  000004          4$:     MOV     R3,@#4          ;RESTORE TIMEOUT VECTOR


                                ;*************************************************************************
                                ;TEST 640       TEST THAT TCSR2 BIT7(DONE) CLEARS WHEN XBUF IS LOADED
                                ;*************************************************************************
124606                          TS640:
124606  032737  000001  001020          BIT     #1, @#$ENV      ;ARE WE RUNNING UNDER APT
124614  001403                          BEQ     70$             ;IF NO THEN DO TEST
124616  005737  001006                  TST     @#$PASS         ;IS THIS FIRST PASS
124622  001022                          BNE     TS641           ;IF NO THEN SHIP TO NEXT TEST
124624                          70$:
124624  005077  177640                  CLR     @TBUF2          ;LOAD XBUF
124630  105777  177632                  TSTB    @TCSR2          ;CHECK DONE
124634  100006                          BPL     3$              ;BR IF CLEAR
                                                                ;FILL SECOND BUFFER BECUASE REFRESH COULD CAUSE
                                                                ;FIRST TEST TO FAIL
124636  005077  177626                  CLR     @TBUF2          ;FILL DOUBLE BUFFER
124642  105777  177620                  TSTB    @TCSR2          ;CHECK DONE
124646  100001                          BPL     3$
124650  104000                          EMT                     ;
124652  005000                  3$:     CLR     R0              ;CLEAR TIMER
124654  105777  177606          4$:     TSTB    @TCSR2          ;CHECK FOR XMIT DONE
124660  100403                          BMI     5$              ;IF DONE SETS, BR TO END OF TEST
124662  005200                          INC     R0              ;INCREMENT TIMER
124664  001373                          BNE     4$
124666  104000                          EMT                     ;
124670                          5$:


                                ;*************************************************************************
                                ;TEST 641       TEST THAT TCSR2 "DONE" SETS WITH RESET
                                ;*************************************************************************
124670                          TS641:
124670  032737  000001  001020          BIT     #1, @#$ENV      ;ARE WE RUNNING UNDER APT
124676  001403                          BEQ     70$             ;IF NO THEN DO TEST
124700  005737  001006                  TST     @#$PASS         ;IS THIS FIRST PASS
124704  001015                          BNE     TS642           ;IF NO THEN SHIP TO NEXT TEST
124706                          70$:
124706  005077  177556                  CLR     @TBUF2          ;LOAD TRANSMIT BUFFER
124712  105777  177550          1$:     TSTB    @TCSR2          ;WAIT FOR DONE
124716  100375                          BPL     1$
124720  005077  177544                  CLR     @TBUF2          ;LOAD SECOND BUFFER
124724  000240                          NOP
124726  000005                          RESET                   ;SET DONE WITH RESET
124730  105777  177532                  TSTB    @TCSR2          ;CHECK FOR DONE SET
124734  100401                          BMI     TS642
124736  104000                          EMT                     ;



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

```
                                        ;TEST 642       TEST ABILITY TO ACCESS RCSR2
                                        ;*************************************************************************
124740                                  TS642:
124740  013703  000004                          MOV     @#4,R3          ;SAVE TIMEOUT VECTOR
124744  012737  124760  000004                  MOV     #1$,@#4         ;SET UP TIMEOUT VECTOR
124752  005777  177504                          TST     @RCSR2          ;ACCESS RCSR
124756  000401                                  BR      2$
124760                                  1$:
124760  104000                                  EMT                     ;
124762  010337  000004                  2$:     MOV     R3,@#4          ;RESTORE TIMEOUT VECTOR


                                        ;*************************************************************************
                                        ;TEST 643       TEST ABILITY TO ACCESS RBUF2
                                        ;*************************************************************************
124766                                  TS643:
124766  013703  000004                          MOV     @#4,R3          ;SAVE TIMEOUT VECTOR
124772  012737  125006  000004                  MOV     #1$,@#4         ;SET UP TIMEOUT VECTOR
125000  005777  177460                          TST     @RBUF2          ;ACCESS RBUF
125004  000401                                  BR      2$
125006                                  1$:
125006  104000                                  EMT                     ;
125010  010337  000004                  2$:     MOV     R3,@#4          ;RESTORE TIMEOUT VECTOR






                                        ;*************************************************************************
                                        ;TEST 644       TEST THAT BITO(BREAK BIT) CAN BE SET & CLEARED & RESET
                                        ;*************************************************************************
125014                                  TS644:
125014  032777  000001  177444                  BIT     #BITO,@TCSR2    ;CHECK BITO OF TCSR CLEAR
125022  001401                                  BEQ     3$
125024  104000                                  EMT                     ;
125026  052777  000001  177432          3$:     BIS     #BITO,@TCSR2    ;SET BITO IN TCSR
125034  032777  000001  177424                  BIT     #BITO,@TCSR2    ;TEST BITO OF TCSR
125042  001001                                  BNE     4$
125044  104000                                  EMT                     ;
125046  042777  000001  177412          4$:     BIC     #BITO,@TCSR2    ;CLEAR BITO OF TCSR
125054  032777  000001  177404                  BIT     #BITO,@TCSR2    ;TEST BITO OF TCSR
125062  001401                                  BEQ     7$
125064  104000                                  EMT                     ;
125066                                  7$:
125066  032737  000001  001020                  BIT     #1, @#$ENV      ;ARE WE RUNNING UNDER APT
125074  001403                                  BEQ     70$             ;IF NO THEN DO TEST
125076  005737  001006                          TST     @#$PASS         ;IS THIS FIRST PASS
125102  001011                                  BNE     TS645           ;IF NO THEN SHIP TO NEXT TEST
125104                                  70$:
125104  052777  000001  177354                  BIS     #BITO,@TCSR2    ;SET BITO IN TCSR
125112  000005                                  RESET                   ;CLEAR BITO WITH RESET
125114  032777  000001  177344                  BIT     #BITO,@TCSR2    ;TEST BITO CLEAR
125122  001401                                  BEQ     TS645
125124  104000                                  EMT                     ;
```

```

                                        ;*****************************************************************************
                                        ;TEST 645        TEST THAT BIT6(XMIT INT EN) CAN BE SET & RESET
                                        ;*****************************************************************************
125126                                  TS645:
125126  017703  177344                          MOV     @TVECT2,R3          ;SAVE XMIT VECTOR
125132  012777  125154  177336                  MOV     #1$,@TVECT2         ;SET UP INTERRUPT VECTOR FOR ERROR REPORT
125140  106427  000340                          MTPS    #340                ;SET PSW TO PRIORITY 7
125144  032777  000100  177314                  BIT     #BIT6,@TCSR2        ;TEST BIT6 OF TCSR
125152  001401                                  BEQ     2$
125154                                  1$:
125154  104000                                  EMT                         ;
125156  052777  000100  177302          2$:     BIS     #BIT6,@TCSR2        ;SET BIT6 OF TCSR
125164  032777  000100  177274                  BIT     #BIT6,@TCSR2        ;TEST BIT6 OF TCSR
125172  001001                                  BNE     3$
125174  104000                                  EMT                         ;
125176  042777  000100  177262          3$:     BIC     #BIT6,@TCSR2        ;CLEAR BIT6 OF TCSR
125204  032777  000100  177254                  BIT     #BIT6,@TCSR2        ;TEST BIT6 OF TCSR
125212  001401                                  BEQ     4$
125214  104000                                  EMT                         ;
125216  032737  000001  001020          4$:     BIT     #1,@#$ENV           ;ARE WE RUNNING UNDER APT
125224  001403                                  BEQ     70$                 ;IF NO THEN DO TEST
125226  005737  001006                          TST     @#$PASS             ;IF THIS FIRST PASS
125232  001011                                  BNE     5$                  ;IF NO THEN SKIP TO END OF TEST
125234                                  70$:
125234  052777  000100  177224                  BIS     #BIT6,@TCSR2        ;SET BIT6 OF TCSR
125242  000005                                  RESET                       ;CLEAR BIT6 WITH RESET
125244  032777  000100  177214                  BIT     #BIT6,@TCSR2        ;TEST BIT6 OF TCSR
125252  001401                                  BEQ     5$
125254  104000                                  EMT                         ;
125256  010377  177214                  5$:     MOV     R3,@TVECT2          ;RESTORE XMIT VECTOR


                                        ;*****************************************************************************
                                        ;TEST 646        TEST THAT BIT6 OF RCSR2 CAN BE SET & RESET
                                        ;*****************************************************************************
125262                                  TS646:
125262  017703  177204                          MOV     @RVECT2,R3          ;SAVE RECEIVE VECTOR
125266  012777  125310  177176                  MOV     #1$,@RVECT2         ;SET UP INTERRUPT VECTOR FOR ERROR REPORT
125274  106427  000340                          MTPS    #340                ;SET PSW TO PRIORITY 7
125300  032777  000100  177154                  BIT     #BIT6,@RCSR2        ;TEST BIT6 OF RCSR
125306  001401                                  BEQ     2$
125310                                  1$:
125310  104000                                  EMT                         ;
125312  052777  000100  177142          2$:     BIS     #BIT6,@RCSR2        ;SET BIT6 OF RCSR
125320  032777  000100  177134                  BIT     #BIT6,@RCSR2        ;TEST BIT6 OF RCSR
125326  001001                                  BNE     3$
125330  104000                                  EMT                         ;
125332  042777  000100  177122          3$:     BIC     #BIT6,@RCSR2        ;CLEAR BIT6 OF RCSR
125340  032777  000100  177114                  BIT     #BIT6,@RCSR2        ;TEST BIT6 OF RCSR
125346  001401                                  BEQ     4$
125350  104000                                  EMT                         ;
125352  032737  000001  001020          4$:     BIT     #1,@#$ENV           ;ARE WE RUNNING UNDER APT
125360  001403                                  BEQ     70$                 ;IF NO THEN DO TEST
125362  005737  001006                          TST     @#$PASS             ;IS THIS FIRST PASS
125366  001011                                  BNE     5$                  ;IF NO THEN SKIP TO END OF TEST
```

```
125370                              70$:
125370  052777  000100  177064              BIS     #BIT6,@RCSR2        ;SET BIT6 OF RCSR
125376  000005                              RESET                       ;CLEAR BIT6OF RCSR2 WITH RESET
125400  032777  000100  177054              BIT     #BIT6,@RCSR2        ;TEST BIT6 OF RCSR
125406  001401                              BEQ     5$
125410  104000                              EMT                         ;
125412  010377  177054              5$:     MOV     R3,@RVECT2          ;RESTORE RECEIVE VECTOR



                                    ;*****************************************************************************
                                    ;TEST 647       TEST THAT XMIT INTERRUPTS ONLY WHEN ENABLED
                                    ;*****************************************************************************
125416                              TS647:
125416  042777  000100  177042              BIC     #BIT6,@TCSR2        ;CLEAR TRANSMIT INTERRUPT ENABLE
125424  017703  177046                      MOV     @TVECT2,R3          ;SAVE XMIT VECTOR
125430  012777  125452  177040              MOV     #2$,@TVECT2         ;POINT XMIT VECTOR TO ERROR REPORT
125436  105777  177024              1$:     TSTB    @TCSR2              ;WAIT FOR DONE
125442  100375                              BPL     1$
125444  106427  000140                      MTPS    #140                ;SET PSW TO PRIORITY 3
125450  000401                              BR      3$
125452                              2$:
125452  104000                              EMT                         ;
125454  012777  125474  177014      3$:     MOV     #4$,@TVECT2         ;SET XMIT VECTOR TO END OF TEST
125462  052777  000100  176776              BIS     #BIT6,@TCSR2        ;ENABLE INTERRUPTS
125470  000240                              NOP

125472  104000                              EMT                         ;XMIT DID NOT INTERRUPT

125474  042777  000100  176764      4$:     BIC     #BIT6,@TCSR2        ;DISABLE INTERRUPTS
125502  022626                              CMP     (SP)+,(SP)+         ;RESTORE SP AFTER INTERRUPT
125504  010377  176766                      MOV     R3,@TVECT2          ;RESTORE XMIT VECTOR


                                    ;*****************************************************************************
                                    ;TEST 650       TEST THAT XMIT INTERRUPTS DO NOT OCCUR WHEN DISABLED
                                    ;*****************************************************************************
125510                              TS650:
125510  042777  000100  176750              BIC     #BIT6,@TCSR2        ;DISABLE INTERRUPTS
125516  106427  000340                      MTPS    #340                ;SET PSW TO PRIORITY 7
125522  017703  176750                      MOV     @TVECT2,R3          ;SAVE XMIT VECTOR
125526  012777  125554  176742              MOV     #2$,@TVECT2         ;POINT XMIT VECTOR TO ERROR REPORT
125534  105777  176726              1$:     TSTB    @TCSR2              ;WAIT FOR DONE
125540  100375                              BPL     1$
125542  052777  000100  176716              BIS     #BIT6,@TCSR2        ;ENABLE INTERRUPT
125550  000240                              NOP
125552  000401                              BR      3$
125554                              2$:
125554  104000                              EMT                         ;
125556  042777  000100  176702      3$:     BIC     #BIT6,@TCSR2        ;CLEAR INTERRUPT ENABLE
125564  012777  125602  176704              MOV     #4$,@TVECT2         ;POINT XMIT VECTOR TO ERROR REPORT
125572  106427  000140                      MTPS    #140                ;SET PSW TO PRIORITY 3
125576  000240                              NOP
125600  000401                              BR      5$
125602                              4$:
125602  104000                              EMT                         ;
```

```
125604 010377 176666          5$:     MOV     R3,@TVECT2        ;RESTORE XMIT VECTOR


                              ;*************************************************************************
                              ;TEST 651       TEST TRANSMITTER FOR DOUBLE INTERRUPTS
                              ;*************************************************************************
125610                        TS651:
125610 042777 000100 176650           BIC     #BIT6,@TCSR2      ;CLEAR INTERRUPT ENABLE
125616 017703 176654                  MOV     @TVECT2,R3        ;SAVE XMIT VECTOR
125622 017704 176652                  MOV     @TPSW2,R4         ;SAVE XMIT PSW VECTOR
125626 012777 125666 176642           MOV     #2$,@TVECT2       ;SET UP XMIT VECTOR
125634 012777 000340 176636           MOV     #340,@TPSW2       ;SET PIO 7 AFTER INTERRUPT
125642 106427 000140                  MTPS    #140              ;SET PSW TO PRIORITY 3
125646 105777 176614          1$:     TSTB    @TCSR2            ;WAIT FOR DONE
125652 100375                         BPL     1$
125654 052777 000100 176604           BIS     #BIT6,@TCSR2      ;ENABLE INTERRUPTS
125662 000240                         NOP

125664 104000                         EMT                       ;
                                                                ;XMIT INTERRUPT DID NOT OCCUR
125666 022626                 2$:     CMP     (SP)+,(SP)+       ;RESTORE SP AFTER INTERRUPT
125670 012777 125714 176600           MOV     #4$,@TVECT2       ;POINT XMIT VECTOR TO ERROR
125676 106427 000140                  MTPS    #140              ;SET PSW TO PRIORITY 3
125702 000240                         NOP                       ;GIVE TIME FOR ANY INTERRUPTS
125704 042777 000100 176554           BIC     #BIT6,@TCSR2      ;DISABLE INTERRUPTS
125712 000401                         BR      5$
125714                        4$:
125714 104000                         EMT                       ;
125716 010377 176554          5$:     MOV     R3,@TVECT2        ;RESTORE XMIT VECTOR
125722 010477 176552                  MOV     R4,@TPSW2         ;RESTORE XMIT PSW VECTOR

                              ;*************************************************************************
                              ;TEST 652       TEST THAT XMIT INTERRUPT CLEARS WITH LOADING TBUF2
                              ;*************************************************************************
125726                        TS652:
125726 032737 000001 001020           BIT     #1, @#$ENV        ;ARE WE RUNNING UNDER APT
125734 001403                         BEQ     70$               ;IF NO THEN DO TEST
125736 005737 001006                  TST     @#$PASS           ;IS THIS FIRST PASS
125742 001043                         BNE     TS653             ;IF NO THEN SHIP TO NEXT TEST
125744                        70$:
125744 042777 000100 176514           BIC     #BIT6,@TCSR2      ;DISABLE INTERRUPTS
125752 106427 000340                  MTPS    #340              ;SET PSW TO PRIORITY 7
125756 017703 176514                  MOV     @TVECT2,R3        ;SAVE XMIT VECTOR
125762 012777 126032 176506           MOV     #2$,@TVECT2       ;POINT XMIT VECTOR TO ERROR
125770 052777 000100 176470           BIS     #BIT6,@TCSR2      ;ENABLE INTERRUPTS
125776 005077 176466                  CLR     @TBUF2            ;LOAD TBUF
126002 105777 176460          1$:     TSTB    @TCSR2            ;WAIT FOR DONE (INTERRUPT)
126006 100375                         BPL     1$
126010 005077 176454                  CLR     @TBUF2            ;FILL SECOND BUFFER TO RESET INT.
126014 106427 000140                  MTPS    #140              ;SET PSW TO PRIORITY 3
126020 000240                         NOP                       ;GIVE TIME FOR ANY INTERRUPTS
126022 042777 000100 176436           BIC     #BIT6,@TCSR2      ;DISABLE INTERRUPTS
126030 000401                         BR      3$
126032                        2$:
126032 104000                         EMT                       ;
126034 010377 176436          3$:     MOV     R3,@TVECT2        ;RESTORE XMIT VECTOR
```

```
126040  005000                          CLR     R0              ;INIT LOOP COUNTER
126042  005200                  4$:     INC     R0              ;INCREMENT COUNTER
126044  001376                          BNE     4$              ;UNTIL COUNTER = 0
126046  005777  176412                  TST     @RBUF2          ;CLEAR RECEIVER BUFFER


                                ;*************************************************************************
                                ;TEST 653       TEST THAT RCVR DONE (7) SET & CLEAR PROPERLY
                                ;*************************************************************************
126052                          TS653:
126052  005000                          CLR     R0              ;CLEAR A TIMER
126054  005077  176410                  CLR     @TBUF2          ;LOAD TRANSMIT BUFFER
126060  105777  176376          WDONE2: TSTB    @RCSR2          ;CHECK FOR RECEIVER DONE
126064  100403                          BMI     6$              ;BR, IF DONE
126066  005200                          INC     R0              ;INCREMENT TIMER, IF NOT DONE
126070  001373                          BNE     WDONE2
126072  104000                          EMT                     ;RECEIVER DONE NEVER SET

126074  032737  000001  001020  6$:     BIT     #1,@#$ENV       ;ARE WE RUNNING UNDER APT
126102  001403                          BEQ     70$             ;IF NO THEN DO TEST
126104  005737  001006                  TST     @#$PASS         ;IS THIS FIRST PASS
126110  001005                          BNE     2$              ;IF NO THEN SKIP TO END OF TEST
126112                          70$:
126112  000005                          RESET                   ;CLEAR DONE WITH RESET
126114  105777  176342                  TSTB    @RCSR2          ;CHECK FOR DONE CLEAR
126120  001401                          BEQ     2$
126122  104000                          EMT                     ;RESET DID NOT CLEAR RCVR DONE
126124  005000                  2$:     CLR     R0              ;INIT LOOP COUNTER
126126  005200                  3$:     INC     R0              ;INCREMENT COUNTER
126130  001376                          BNE     3$              ;UNTIL COUNTER = 0
126132  005777  176326                  TST     @RBUF2          ;CLEAR RECEIVER BUFFER


                                ;*************************************************************************
                                ;TEST 654       TEST THAT READING RBUF2 CLEARS RECEIVER DONE
                                ;*************************************************************************
126136                          TS654:
126136  005077  176326                  CLR     @TBUF2          ;LOAD TRANSMITTER
126142  105777  176314          1$:     TSTB    @RCSR2          ;WAIT FOR RECEIVER DONE
126146  100375                          BPL     1$
126150  017700  176310                  MOV     @RBUF2,R0       ;READ RECEIVE BUFFER
126154  105777  176302                  TSTB    @RCSR2          ;CHECK FOR RECEIVE DONE CLEAR
126160  001401                          BEQ     TS655
126162  104000                          EMT                     ;
                                                                ;READING RBUF2 DID NOT CLEAR RCVR DONE


                                ;*************************************************************************
                                ;TEST 655       TEST THAT RCVR INTERRUPTS ONLY WHEN ENABLED
                                ;*************************************************************************
126164                          TS655:
126164  042777  000100  176274          BIC     #BIT6,@TCSR2    ;DISABLE TRANSMIT INTERRUPTS
126172  042777  000100  176262          BIC     #BIT6,@RCSR2    ;DISABLE RECEIVER INTERRUPTS
126200  017703  176266                  MOV     @RVECT2,R3      ;SAVE RECEIVE VECTOR
126204  012777  126232  176260          MOV     #2$,@RVECT2     ;POINT RCV VECTOR TO ERROR REPORT
126212  106427  000140                  MTPS    #140            ;SET PSW TO PRIORITY 3
```

```
126216  005077  176246                  CLR     @TBUF2          ;SEND A CHARACTER
126222  105777  176234          1$:     TSTB    @RCSR2          ;WAIT FOR RECEIVER DONE
126226  100375                          BPL     1$
126230  000401                          BR      3$
126232                          2$:
126232  104000                          EMT                     ;
126234  012777  126254  176230  3$:     MOV     #4$,@RVECT2     ;POINT RCV VECTOR TO END OF TEST
126242  052777  000100  176212          BIS     #BIT6,@RCSR2    ;ENABLE RCV INTERRUPTS
126250  000240                          NOP                     ;GIVE ANY INTERRUPTS TIME
126252  104000                          EMT                     ;
126254  042777  000100  176200  4$:     BIC     #BIT6,@RCSR2    ;DISABLE INTERRUPTS
126262  022626                          CMP     (SP)+,(SP)+     ;RESTORE SP AFTER INTERRUPT
126264  005777  176174                  TST     @RBUF2          ;CLEAR CHARACTER FROM RECEIVER BUFFER
126270  010377  176176                  MOV     R3,@RVECT2      ;RESTORE RECEIVE VECTOR


                                ;*****************************************************************************
                                ;TEST 656       TEST THAT RCVR INTERRUPTS DO NOT OCCUR WHEN DISABLED
                                ;*****************************************************************************
126274                          TS656:
126274  106427  000340                  MTPS    #340            ;SET PSW TO PRIORITY 7
126300  017703  176166                  MOV     @RVECT2,R3      ;SAVE RECEIVE VECTOR
126304  012777  126336  176160          MOV     #2$,@RVECT2     ;POINT RCVR VECTOR TO ERROR REPORT
126312  005077  176152                  CLR     @TBUF2          ;SEND A CHARACTER
126316  105777  176140          1$:     TSTB    @RCSR2          ;WAIT FOR RECEIVER DONE
126322  100375                          BPL     1$
126324  052777  000100  176130          BIS     #BIT6,@RCSR2    ;ENABLE INTERRUPTS
126332  000240                          NOP                     ;GIVE TIME FOR INTERRUPT
126334  000401                          BR      3$
126336                          2$:
126336  104000                          EMT                     ;RCVR INTERRUPTS AT PRIORITY 7
126340  042777  000100  176114  3$:     BIC     #BIT6,@RCSR2    ;CLEAR INTERRUPT ENABLE
126346  012777  126364  176116          MOV     #4$,@RVECT2     ;POINT RCVR VECTOR TO ERROR REPORT
126354  106427  000140                  MTPS    #140            ;SET PSW TO PRIORITY 3
126360  000240                          NOP                     ;GIVE TIME FOR ANY INTERRUPT
126362  000401                          BR      5$
126364                          4$:
126364  104000                          EMT                     ;RCVR INTERRUPT REQUEST PASSED WITH BIT6 CLEAR
126366  005777  176072          5$:     TST     @RBUF2          ;CLEAR CHARACTER FROM RECEIVER BUFFER
126372  010377  176074                  MOV     R3,@RVECT2      ;RESTORE RECEIVE VECTOR


                                ;*****************************************************************************
                                ;TEST 657       TEST RECEIVER FOR DOUBLE INTERRUPTS
                                ;*****************************************************************************
126376                          TS657:
126376  017703  176070                  MOV     @RVECT2,R3      ;SAVE RECEIVE VECTOR
126402  017704  176066                  MOV     @RPSW2,R4       ;SAVE RECEIVE PSW VECTOR
126406  012777  126452  176056          MOV     #2$,@RVECT2     ;POINT RCV VECTOR TO CONTINUE TEST
126414  012777  000340  176052          MOV     #340,@RPSW2     ;SET PRIORITY TO 7 AFTER INTERRUPT
126422  106427  000140                  MTPS    #140            ;SET PSW TO PRIORITY 3
126426  005077  176036                  CLR     @TBUF2          ;SEND A CHARACTER
126432  105777  176024          1$:     TSTB    @RCSR2          ;WAIT FOR RCVR DONE
126436  100375                          BPL     1$
126440  052777  000100  176014          BIS     #BIT6,@RCSR2    ;ENABLE RCV INTERRUPTS
126446  000240                          NOP                     ;GIVE SOME TIME
```

```
126450 104000                          EMT                          ;
126452 022626                  2$:     CMP     (SP)+,(SP)+          ;RESTORE SP AFTER INTERRUPT
126454 012777 126510 176010            MOV     #3$,@RVECT2          ;POINT RCV VECTOR TO ERROR REPORT
126462 106427 000140                   MTPS    #140                 ;SET PSW TO PRIORITY 3
126466 000240                          NOP                          ;GIVE SOME TIME
126470 042777 000100 175764            BIC     #BIT6,@RCSR2         ;CLEAR INTERRUPT ENABLE
126476 010377 175770                   MOV     R3,@RVECT2           ;RESTORE RECEIVE VECTOR
126502 010477 175766                   MOV     R4,@RPSW2            ;RESTORE RECEIVE PSW VECTOR
126506 000401                          BR      4$
126510                         3$:
126510 104000                          EMT                          ;
126512 005777 175746           4$:     TST     @RBUF2               ;CLEAR CHARACTER FROM RECEIVER BUFFER
126516 010377 175750                   MOV     R3,@RVECT2           ;RESTORE RECEIVE VECTOR



                               ;*************************************************************************
                               ;TEST 660        TEST THAT RCVR INTERRUPT CLEARS BY READING RBUF2
                               ;*************************************************************************
126522                         TS660:
126522 106427 000340                   MTPS    #340                 ;SET PSW TO PRIORITY 7
126526 017703 175740                   MOV     @RVECT2,R3           ;SAVE RECEIVE VECTOR
126532 012777 126602 175732            MOV     #2$,@RVECT2          ;POINT RCV VECTOR TO ERROR REPORT
126540 052777 000100 175714            BIS     #BIT6,@RCSR2         ;SET RCVR INTERRUPT ENABLE
126546 005077 175716                   CLR     @TBUF2               ;SEND A CHARACTER
126552 105777 175704           1$:     TSTB    @RCSR2               ;WAIT FOR DONE (INTERRUPT)
126556 100375                          BPL     1$
126560 005777 175700                   TST     @RBUF2               ;READ RBUF TO CLEAR PENDING INTERRPUT
126564 106427 000140                   MTPS    #140                 ;SET PSW TO PRIORITY 3
126570 000240                          NOP                          ;ALLOW TIME FOR ANY ERRONEOUS INTERRUPT
126572 042777 000100 175662            BIC     #BIT6,@RCSR2         ;NO INTERRUPT-CLEAR INT. ENABLE
126600 000401                          BR      3$
126602                         2$:
126602 104000                          EMT                          ;
126604 010377 175662           3$:     MOV     R3,@RVECT2           ;RESTORE RECEIVE VECTOR



                               ;*************************************************************************
                               ;TEST 661        TEST THAT RESET CLEARS RECEIVE INTERRUPT
                               ;*************************************************************************
126610                         TS661:
126610 032737 000001 001020            BIT     #1, @#$ENV           ;ARE WE RUNNING UNDER APT
126616 001403                          BEQ     70$                  ;IF NO THEN DO TEST
126620 005737 001006                   TST     @#$PASS              ;IS THIS FIRST PASS
126624 001036                          BNE     TS662                ;IF NO THEN SHIP TO NEXT TEST
126626                         70$:
126626 106427 000340                   MTPS    #340                 ;SET PSW TO PRIORITY 7
126632 017703 175634                   MOV     @RVECT2,R3           ;SAVE RECEIVE VECTOR
126636 012777 126714 175626            MOV     #2$,@RVECT2          ;POINT RCV VECTOR TO ERROR REPORT
126644 052777 000100 175610            BIS     #BIT6,@RCSR2         ;SET RCV INTERRUPT ENABLE
126652 012777 000377 175610            MOV     #377,@TBUF2          ;SEND AN ALL 1'S CHARACTER
126660 105777 175576           1$:     TSTB    @RCSR2               ;WAIT FOR RCV DONE
126664 100375                          BPL     1$
126666 000005                          RESET                        ;CLEAR RCV INTERRUPT & RBUF2
126670 052777 000100 175564            BIS     #BIT6,@RCSR2         ;SET RECEIVER INTERRUPT
126676 106427 000140                   MTPS    #140                 ;SET PSW TO PRIORITY 3
```

```
126702  000240                                  NOP                             ;ALLOW TIME FOR AN ERRONEOUS INTERRUPT
126704  042777  000100  175550                  BIC     #BIT6,@RCSR2            ;NO INTERRUPT-CLEAR INT. ENABLE
126712  000401                                  BR      3$
126714                                  2$:
126714  104000                                  EMT                             ;
126716  010377  175550                  3$:     MOV     R3,@RVECT2              ;RESTORE RECEIVE VECTOR


                                        ;*******************************************************************************
                                        ;TEST 662       TEST THAT THE ''OR'' ERROR (BIT14) & ''ERROR'' (BIT15) CAN BE SET
                                        ;*******************************************************************************
126722                                  TS662:
126722  012700  000003                          MOV     #3,R0                   ;SET CHARACTER COUNT TO SEND 3 CHAR.
126726  005077  175536                  1$:     CLR     @TBUF2                  ;LOAD TRANSMIT BUFFER
126732  105777  175530                  2$:     TSTB    @TCSR2                  ;WAIT FOR TRANSMIT DONE
126736  100375                                  BPL     2$
126740  005300                                  DEC     R0                      ;DECREMENT CHARACTER COUNT
126742  001371                                  BNE     1$                      ;BR IF ALL CHARACTERS NOT TRANSMITTED
126744  032777  040000  175512                  BIT     #BIT14,@RBUF2           ;TEST FOR ''OR'' ERROR FLAG
126752  001001                                  BNE     3$
126754  104000                                  EMT                             ;
126756  032777  100000  175500          3$:     BIT     #BIT15,@RBUF2           ;TEST ''ERROR'' FLAG
126764  001001                                  BNE     4$
126766  104000                                  EMT                             ;
126770  005000                          4$:     CLR     R0                      ;CLEAR LOOP COUNTER
126772  005200                          5$:     INC     R0                      ;INCREMENT LOOP COUNTER
126774  001376                                  BNE     5$                      ;  UNTIL COUNTER = 0
126776  005777  175462                          TST     @RBUF2                  ;CLEAR CHARACTER FROM RECEIVER BUFFER


                                        ;*******************************************************************************
                                        ;TEST 663       TEST THAT BREAK TRANSMITS ALL ZEROES
                                        ;*******************************************************************************
127002                                  TS663:
127002  032737  000001  001020                  BIT     #1, @#$ENV              ;ARE WE RUNNING UNDER APT
127010  001403                                  BEQ     70$                     ;IF NO THEN DO TEST
127012  005737  001006                          TST     @#$PASS                 ;IS THIS FIRST PASS
127016  001027                                  BNE     TS664                   ;IF NO THEN SHIP TO NEXT TEST
127020                                  70$:
127020  012777  177777  175442                  MOV     #-1,@TBUF2              ;TRANSMIT ALL ONES TO RCVR
127026  105777  175430                  1$:     TSTB    @RCSR2                  ;WAIT FOR RCVR DONE
127032  100375                                  BPL     1$
127034  005777  175424                          TST     @RBUF2                  ;CLEAR DONE (LEAVING ALL ONES IN RBUF)
127040  052777  000001  175420                  BIS     #BIT0,@TCSR2            ;TRANSMIT BREAK
127046  005000                                  CLR     R0                      ;CLEAR A TIMER
127050  105777  175406                  2$:     TSTB    @RCSR2                  ;WAIT FOR RCVR DONE
127054  100403                                  BMI     CONT42                  ;BR IF DONE
127056  005200                                  INC     R0                      ;IF NOT, INCREMENT TIMER
127060  001373                                  BNE     2$
127062  104000                                  EMT                             ;BREAK DID NOT TRANSMIT ANYTHING

127064  105777  175374                  CONT42: TSTB    @RBUF2                  ;CHECK RECEIVE BUFFER FOR ZERO
127070  001401                                  BEQ     3$
127072  104000                                  EMT                             ;BREAK DID NOT TRANSMIT ALL ZEROES

127074  000005                          3$:     RESET                           ;CLEAR ERRORS
```

```



                                             ;************************************************************************
                                             ;TEST 664        TEST THAT "FR" ERROR CAN BE SET DURING BREAK
                                             ;************************************************************************
127076                                       TS664:
127076  052777  000001  175362                        BIS     #BIT0,@TCSR2      ;SEND BREAK
127104  005077  175360                                CLR     @TBUF2            ;TRANSMIT A CHARACTER TO TIME BREAK
127110  105777  175346                       1$:      TSTB    @RCSR2            ;WAIT FOR RCVR DONE
127114  100375                                        BPL     1$
127116  042777  000001  175342                        BIC     #BIT0,@TCSR2      ;CLEAR BREAK BITS
127124  032777  020000  175332                        BIT     #BIT13,@RBUF2     ;CHECK FOR FRAMING ERROR FLAG
127132  001001                                        BNE     2$
127134  104000                                        EMT                       ;BREAK DID NOT SET FRAMING ERROR
127136  032777  100000  175320               2$:      BIT     #BIT15,@RBUF2     ;TEST "ERROR" FLAG
127144  001001                                        BNE     3$
127146  104000                                        EMT                       ;"ERROR" FLAG DID NOT SET WITH "OR" FLAG
127150  005777  175310                       3$:      TST     @RBUF2            ;CLEAR RECEIVER BUFFER

                                             ;************************************************************************
                                             ;TEST 665        TEST DATA PATHS USING WRAP CABLE
                                             ;************************************************************************
127154                                       TS665:
127154  005001                                        CLR     R1                ;CLEAR REGISTER FOR TEST DATA
127156  105201                               1$:      INCB    R1                ;INCREMENT THE TEST DATA
127160  010177  175304                                MOV     R1,@TBUF2         ;XMIT A CHARACTER
127164  005000                                        CLR     R0                ;CLEAR A TIMER
127166  105777  175270                       2$:      TSTB    @RCSR2            ;WAIT FOR RECEIVER DONE
127172  100403                                        BMI     3$                ;BR IF DONE
127174  005200                                        INC     R0                ;INCREMENT TIMER IF NOT
127176  001373                                        BNE     2$
127200  104000                                        EMT                       ;
127202  017702  175256                       3$:      MOV     @RBUF2,R2         ;GET RECEIVED CHARACTER
127206  020102                                        CMP     R1,R2             ;COMPARE DATA
127210  001401                                        BEQ     4$
127212  104000                                        EMT                       ;
127214  105701                               4$:      TSTB    R1                ;TEST XMIT DATA FOR ZERO
127216  001357                                        BNE     1$                ;BR, IF NOT FINISHED
127220  000167  000102                                JMP     UNIQUE            ;FINISHED TESTING DEVICES SEPARATELY
                                                                                ;GO TEST THEM ALL TOGETHER
127224  000000                               $BDADR: 0
127226  000000                               $BDDAT: 0
127230  000000                               $GDADR: 0
127232  000000                               $GDDAT: 0


127234  012737  127266  000004               ERROR7: MOV     #1$,@#4           ;SET UP THE TIMEOUT VECTOR
127242  012737  000340  000006                        MOV     #340,@#6
127250  005737  177766                                TST     @#177766          ;CHECK TO SEE IF THE POWER FAILED,WILL TRAP
                                                                                ;TO 1$ IF UNIBUS MAP BOARD IS NOT PRESENT
                                                                                ;SINCE THE CPUERR(177766)IS ON THE UNIBUS
                                                                                ;MAP BOARD
127254  001404                                        BEQ     1$                ;NO,GO REPORT THE ERROR
127256  012737  000177  001002                        MOV     #177,@#$FATAL     ;YES,FLAG THAT THE POWER FAILED
127264  000403                                        BR      2$                ; AND GO SET FATAL ERROR FLAG
127266  012737  000006  001002               1$:      MOV     #6,@#$FATAL                 ;SET UP FATAL ERROR NUMBER
```

```
127274  012767  000001  051476  2$:      MOV     #1,$MSGTY               ;SET FATAL ERROR FLAG
127302  000777                  SL2HLT:  BR      .


127304  177560                  DADTBL:  .WORD   177560
127306  177562                           .WORD   177562
127310  177564                           .WORD   177564
127312  177566                           .WORD   177566
127314  176500                           .WORD   176500
127316  176502                           .WORD   176502
127320  176504                           .WORD   176504
127322  176506                           .WORD   176506
127324  177564                  TBLEND:  .WORD   177564


127326  032777  000034  071770  UNIQUE:  BIT     #34,@SWR
127334  001402                           BEQ     1$
127336  000167  002112                   JMP     ENDPAS
127342  012737  000007  001004  1$:      MOV     #7,@#$TESTN     ;UPDATE TEST NUMBER FOR APT
127350  012737  130404  000030           MOV     #ERROR8,@#30    ;SET UP FOR CORRECT ERROR CALL

                                ;*******************************************************************************
                                ;TEST 666        UNIQUE INTERNAL ADDRESS TEST
                                ;*******************************************************************************
127356                          TS666:
127356  032737  000001  001020           BIT     #1, @#$ENV      ;ARE WE RUNNING UNDER APT
127364  001403                           BEQ     70$             ;IF NO THEN DO TEST
127366  005737  001006                   TST     @#$PASS         ;IS THIS FIRST PASS
127372  001044                           BNE     TS667           ;IF NO THEN SHIP TO NEXT TEST
127374                          70$:
127374  012767  000340  050374           MOV     #340,PS         ;WE WILL BE PLAYING WITH BIT6
                                                                 ;SO LOCK OUT EXTRAINEOUS INTERRUPTS
127402  012700  127304                   MOV     #DADTBL,R0      ;GET LOCATION OF FIRST REGISTER ADDRESS
127406  012703  127304          1$:      MOV     #DADTBL,R3      ;MAKE R3 POINT TO LOCATION OF FIRST
                                                                 ;REGISTER ADDRESS
127412  012701  000011                   MOV     #11,R1          ;SET LOOP COUNTER TO CLEAR ALL REG.
127416  005033                  2$:      CLR     @(R3)+          ;CLEAR A REGISTER
127420  077102                           SOB     R1,2$           ;LOOP UNTIL ALL REGISTERS CLEARED
127422  012770  000100  000000           MOV     #BIT6,@(R0)     ;SET TEST BIT IN DEVICE REGISTERS
127430  012701  127304                   MOV     #DADTBL,R1      ;GET LOCATION OF FIRST REGISTER ADDRESS
127434  012702  000011                   MOV     #11,R2          ;SET UP TEST LOOP COUNTER
127440  032731  000100          3$:      BIT     #BIT6,@(R1)+    ;IS TEST BIT SET IN THIS REGISTER
127444  001006                           BNE     5$              ;IF YES GO SEE IF THERE IS AN ERROR
127446  077204                  4$:      SOB     R2,3$           ;LOOP UNTIL ALL REGISTER CHECKED
127450  005030                           CLR     @(R0)+          ;CLEAR REGISTER JUST TESTED AND POINT
                                                                 ;TO NEXT ONE
127452  020027  127324                   CMP     R0,#TBLEND              ;ARE WE DONE TESTING
127456  001407                           BEQ     7$              ;IF YES GO TO NEXT TEST
127460  000752                           BR      1$              ;CONTINUE TESTING
127462  021041                  5$:      CMP     (R0),-(R1)      ;DID WE COMPARE THE REGISTER TO ITSELF?
127464  001401                           BEQ     6$
127466  104000                           EMT                     ;WRITE TO 1 INTERNAL ADDRESS MODIFIED
                                                                 ;ANOTHER SO ADDRESS NOT UNIQUE
127470  062701  000002          6$:      ADD     #2,R1           ;PESTORE POINTER
127474  000764                           BR      4$              ;GET BACK IN TEST LOOP
127476  005000                  7$:      CLR     R0              ;INITIALIZE LOOP COUNTER
```

```
127500 005200                   8$:      INC     R0                 ;INCREMEMT COUNTER
127502 001376                            BNE     8$                 ;  UNTIL LOOP COUNTER = 0
                                ;*************************************************************************
                                ;TEST 667        TEST ALL INTERNAL OPTIONS SIMULTANEOUSLY
                                ;*************************************************************************
127504                          TS667:
127504 032737 000001 001020              BIT     #1, @#$ENV         ;ARE WE RUNNING UNDER APT
127512 001405                            BEQ     70$                ;IF NO DO TEST
127514 005737 001006                     TST     @#$PASS            ;'S THIS FIRST PASS
127520 001402                            BEQ     70$                ;IF YES DO TEST
127522 000167 001726                     JMP     ENDPAS             ;IF NO THEN SKIP THIS TEST
127526 000005                   70$:     RESET                      ;CLEAR EVERY BODY
127530 012767 000340 050240              MOV     #340,PS            ;SET PROCESSOR PRIORITY TO 7
127536 017767 174734 107616              MOV     @TVECT2,$TMP0
127544 017767 174722 107612              MOV     @RVECT2,$TMP1
127552 017767 170770 107606              MOV     @TVECT,$TMP2
127560 017767 170756 107602              MOV     @RVECT,$TMP3
127566 017767 170762 107576              MOV     @RTCVT,$TMP4
127574 005067 000572                     CLR     XMTCT1             ;INITIALIZE COUNTERS
127600 005067 000570                     CLR     XMTCT2
127604 005067 000566                     CLR     RECCT1
127610 005067 000564                     CLR     RECCT2
127614 005067 000562                     CLR     TICKS
127620 012777 130020 170720              MOV     #XMIT1,@TVECT      ;SET UP SLU1 TRANSMIT VECTOR
127626 012777 000340 170714              MOV     #340, @TPSW        ;AND PSW
127634 012777 130054 170700              MOV     #REC1, @RVECT      ;SET UP SLU1 RECEIVER VECTOR
127642 012777 000340 170674              MOV     #340, @RPSW        ;AND PSW
127650 012777 130102 174620              MOV     #XMIT2, @TVECT2    ;SET UP SLU2 TRANSMIT VECTOR
127656 012777 000340 174614              MOV     #340, @TPSW2       ;AND PSW
127664 012777 130136 174600              MOV     #REC2, @RVECT2     ;SET UP SLU2 RECEIVER VECTOR
127672 012777 000340 174574              MOV     #340, @RPSW2       ;AND PSW
127700 012777 130010 170646              MOV     #TICKER, @RTCVT    ;SET UP RTC VECTOR
127706 012777 000340 170642              MOV     #340, @RTCPSW      ;AND PSW
127714 052777 000004 170614              BIS     #BIT2, @TCSR       ;ENABLE SLU1 MAINTENANCE WRAP
127722 052777 000100 170606              BIS     #BIT6, @TCSR       ;ENABLE SLU1 XMIT INTERRUPT
127730 052777 000100 170574              BIS     #BIT6, @RCSR       ;ENABLE SLU1 RECEIVER INTERRUPT
127736 012702 130454                     MOV     #BUF1, R2          ;SET UP RECEIVER BUFFER
127742 052777 000100 174516              BIS     #BIT6, @TCSR2      ;ENABLE SLU2 XMIT INTERRUPT
127750 052777 000100 174504              BIS     #BIT6, @RCSR2      ;ENABLE SLU2 RECEIVER INTERRUPT
127756 012703 131054                     MOV     #BUF2, R3          ;SET UP RECEIVER BUFFER
127762 052777 000100 170562              BIS     #BIT6, @LKS        ;ENABLE RTC INTERRUPTS
127770 012700 177777            3$:      MOV     #-1, R0            ;INITIALIZE DATA FOR SLU1
127774 012701 177777                     MOV     #-1, R1            ;INITIALIZE DATA FOR SLU2
130000 005067 047772                     CLR     PS                 ;DROP PROCESSOR PRIORITY TO 0
130004 000001                   WAITIO:  WAIT                       ;WAIT FOR INTERRUPT
130006 000776                            BR      WAITIO
130010 005267 000366            TICKER:  INC     TICKS              ;UPDATE COUNT
130014 000167 000140                     JMP     IOHAND             ;GO TO INTERRUPT HANDLER
130020 005267 000346            XMIT1:   INC     XMTCT1             ;UPDATE XMIT INTERRUPT COUNT
130024 005200                            INC     R0                 ;UPDATE XMIT DATA
130026 010077 170506                     MOV     R0, @TBUF          ;SEND NEXT CHARACTER
130032 026727 000334 000400              CMP     XMTCT1,#400        ;IF 256 CHARACTERS HAVE NOT BEEN
```

```
130040  002403                           BLT     1$              ;TRANSFERRED CONTINUE
130042  042777  000100  170466           BIC     #100,   @TCSR   ;ELSE NO MORE TRANSMITTER INTERRUPTS
130050  000167  000104          1$:      JMP     IOHAND          ;GO TO INTERRUPT HANDLER

130054  005267  000316          REC1:    INC     RECCT1          ;UPDATE RECEIVER INTERRUPT COUNT
130060  005777  170450                   TST     @RBUF           ;BIT 15 SET IF ANY ERRORS OCCURRED
130064  100002                           BPL     3$              ;IF BIT IS CLEAR NO ERROR
130066  000005                           RESET                   ;CLEAR THE WORLD STOP ALL
                                                                 ;INTERRUPTS
130070  104000                           EMT                     ;RECEIVER STATUS ERROR
130072  117722  170436          3$:      MOVB    @RBUF, (R2)+    ;GET DATA AND STORE IT
130076  000167  000056                   JMP     IOHAND          ;GO TO INTERRUPT HANDLER

130102  005267  000266          XMIT2:   INC     XMTCT2          ;UPDATE XMIT INTERRUPT COUNT
130106  005201                           INC     R1              ;UPDATE XMIT DATA
130110  010177  174354                   MOV     R1, @TBUF2      ;SEND NEXT CHARACTER
130114  026727  000254  000400           CMP     XMTCT2, #400    ;IF 256 CHARACTERS HAVE NOT
130122  002403                           BLT     1$              ;BEEN TRANSFERRED CONTINUE
130124  042777  000100  174334           BIC     #BIT6, @TCSR2   ;ELSE NO MORE XMIT INTERRUPTS
130132  000167  000022          1$:      JMP     IOHAND          ;GO TO INTERRUPT HANDLER

130136  005267  000236          REC2:    INC     RECCT2          ;UPDATE RECEIVER INTERRUPT COUNT
130142  005777  174316                   TST     @RBUF2          ;BIT 15 SETS IF ANY ERRORS OCCURRED
130146  100002                           BPL     3$              ;IF BIT IS CLEAR NO ERRORS
130150  000005                           RESET                   ;CLEAR THE WORLD - STOP ALL
                                                                 ;INTERRUPTS
130152  104000                           EMT                     ;RECEIVER STATUS ERROR
130154  117723  174304          3$:      MOVB    @RBUF2, (R3)+   ;GET DATA AND STORE IT

130160  026727  000216  000074  IOHAND:  CMP     TICKS, #74      ;HAS 1 SEC ELAPSED
130166  001401                           BEQ     1$              ;IF YES STOP TEST
130170  000002                           RTI                     ;RETURN FROM INTERRUPT TO AWAIT NEXT
130172  042777  000100  170336  1$:      BIC     #BIT6, @TCSR    ;IF YES STOP TRANSMISSIONS
130200  042777  000100  174260           BIC     #BIT6, @TCSR2   ;
130206  042777  000100  170336           BIC     #BIT6,  @LKS    ;TURN OFF LINE CLOCK

130214  106427  000000          WAITER:  MTPS    #0              ;LOWER PRIORITY TO ALLOW TIME FOR RECEIVER TO FINISH
130220  012705  140000                   MOV     #-40000,R5      ;SET UP LOOP COUNTER
130224  062705  000001          1$:      ADD     #1, R5          ;DO LOOP UNTIL R5 = 0
130230  001375                           BNE     1$
130232  000005                           RESET                   ;STOP EVERYONE SHOULD BE DONE

130234  026767  000132  000134  CHECK1:  CMP     XMTCT1,RECCT1   ;# OF XMIT INTERRUPTS = REC INTERRUPTS
130242  001401                           BEQ     1$
130244  104000                           EMT                     ;INTERRUPT COMPARISON ERROR
130246  012702  130454          1$:      MOV     #BUF1, R2       ;POINT TO FIRST DATA
130252  005000                           CLR     R0              ;INITIALIZE TO FIRST DATA XMIT
130254  016704  000112                   MOV     XMTCT1, R4      ;GET # OF BYTES TRANSFERRED
130260  122200                  2$:      CMPB    (R2)+, R0       ;IS RECEIVED DATA = EXPECTED
130262  001401                           BEQ     3$
130264  104000                           EMT                     ;SLU1 DATA COMPARISON ERROR
130266  005200                  3$:      INC     R0              ;UPDATE TO NEXT GOOD DATA
130270  077405                           SOB     R4,2$           ;LOOP UNTIL ALL DATA CHECKED
130272  026767  000076  000100  CHECK2:  CMP     XMTCT2, RECCT2  ;#OF XMIT INTERRUPTS = REC INTERRUPTS
130300  001401                           BEQ     1$
130302  104000                           EMT                     ;INTERRUPT COMPARISON ERROR
```

```
130304  012703  131054          1$:      MOV     #BUF2, R3          ;INITIALIZE TO FIRST RECEIVED DATA
130310  005001                           CLR     R1                 ;INITIALIZE TO FIRST XMIT DATA
130312  016704  000056                   MOV     XMTCT2, R4         ;GET # OF BYTES TRANSFERRED
130316  122301                  2$:      CMPB    (R3)+, R1          ;IS RECEIVED DATA = EXPECTED DATA
130320  001401                           BEQ     3$
130322  104000                           EMT                        ;SLU2 DATA COMPARISON ERROR
130324  005201                  3$:      INC     R1                 ;UPDATE TO NEXT GOOD DATA
130326  077405                           SOB     R4,2$              ;LOOP UNTIL ALL DATA CHECKED
130330  016777  107026  174140  FINIE:   MOV     $TMP0, @TVECT2     ;RESTORE VECTORS
130336  016777  107022  174126           MOV     $TMP1, @RVECT2
130344  016777  107016  170174           MOV     $TMP2, @TVECT
130352  016777  107012  170162           MOV     $TMP3, @RVECT
130360  016777  107006  170166           MOV     $TMP4, @RTCVT
130366  000167  001062                   JMP     ENDPAS             ;FINISHED TESTING GO TO END OF PASS

130372  000000                  XMTCT1:  .WORD   0
130374  000000                  XMTCT2:  .WORD   0
130376  000000                  RECCT1:  .WORD   0
130400  000000                  RECCT2:  .WORD   0
130402  000000                  TICKS:   .WORD   0

130404  012737  130436  000004  ERROR8:  MOV     #1$,@#4            ;SET UP THE TIMEOUT VECTOR
130412  012737  000340  000006           MOV     #340,@#6
130420  005737  177766                   TST     @#177766           ;CHECK TO SEE IF THE POWER FAILED,WILL TRAP
                                                                    ;TO 1$ IF UNIBUS MAP BOARD IS NOT PRESENT
                                                                    ;SINCE THE CPUERR(177766)IS ON THE UNIBUS
                                                                    ;MAP BOARD
130424  001404                           BEQ     1$                 ;NO,GO REPORT THE ERROR
130426  012737  000177  001002           MOV     #177,@#$FATAL      ;YES,FLAG THAT THE POWER FAILED
130434  000403                           BR      2$                 ; AND GO SET FATAL ERROR FLAG
130436  012737  000007  001002  1$:      MOV     #7,@#$FATAL                        ;SET UP FATAL ERROR NUMBER
130444  012767  000001  050326  2$:      MOV     #1,$MSGTY                          ;SET FATAL ERROR FLAG
130452  000777                  COMHLT:  BR      .

130454  000200                  BUF1:    .BLKW   200
131054  000200                  BUF2:    .BLKW   200




131454  005327                  ENDPAS:  DEC     (PC)+              ;DECREMENT TEST LOOP COUNTER
131456  000001                  $EOPCT:  .WORD   1
131460  003043                           BGT     $DOAGN             ;IF COUNTER NOT 0 DO TEST AGAIN
131462  005267  047320                   INC     $PASS              ;INCREMENT PASS COUNTER
131466  042767  100000  047312           BIC     #100000,$PASS      ;DON'T LET IT BE NEGATIVE
131474  016767  047324  177754           MOV     $USWR,$EOPCT       ;RESET TEST LOOP COUNTER
131502  012700  131713                   MOV     #ENDMSG,R0         ;LET R0 POINT TO ENDPASS MESSAGE
131506  004767  000126                   JSR     PC,TYPE            ;GO TYPE END PASS MESSAGE
131512  016700  046324                   MOV     42,R0              ;GET MONITOR ADDRESS
131516  001405                           BEQ     DOAGIN             ;IF = 0 NO MONITOR SO DON'T STOP
131520  000005                           RESET                      ;IF MONITOR CLEAR THE WORLD
131522  004710                  $ENDAD:  JSR     PC,(R0)            ;GO TO MONITOR
131524  000240                           NOP                        ;THESE THREE LOCATIONS RESERVED
131526  000240                           NOP
131530  000240                           NOP
131532  013737  000004  037362  DOAGIN:  MOV     @#4,@#$TMP0
```

```
131540  012737  131556  000004           MOV     #1$,@#4
131546  012737  000001  164000           MOV     #1,@#164000
131554  000402                           BR      2$
131556  062706  000004          1$:      ADD     #4,SP
131562  013737  037362  000004  2$:      MOV     @#$TMP0,@#4
131570  000137                  $DOAGN:  JMP     @(PC)+          ;RETURN TO TEST AT LOCATION RESTRT
131572  001206                           .WORD   RESTRT

                                ;*******************************************************************
                                ;*COMMON SUBROUTINES THAT ARE NEEDED BY THE PROGRAM
                                ;*
                                ;*******************************************************************

131574  012737  131604  000024  PWRDN:   MOV     #PWRUP,@#24     ;SET UP POWER FAIL VECTOR FOR POWER UP
131602  000000                           HALT

131604  012737  131574  000024  PWRUP:   MOV     #PWRDN,@#24     ;SET UP POWER FAIL VECTOR FOR POWER DOWN
131612  012706  001000                   MOV     #STBOT,SP       ;SET UP STACK
131616  005737  000172                   TST     @#MTFLAG        ;ARE WE ON MULTI-OPTION TESTER
131622  001004                           BNE     1$              ;IF YES SKIP TYPE OUT
131624  012700  131674                   MOV     #PWRMSG,R0      ;POINT R0 TO POWER FAIL MESSAGE
131630  004767  000004                   JSR     PC,TYPE         ;GO TYPE IT
131634  000167  047346          1$:      JMP     RESTRT          ;GO RESTART TEST

131640  132767  000040  047153  TYPE:    BITB    #40,$ENVM       ;TYPE OUTS DISABLED
131646  001011                           BNE     3$              ;IF YES GET TO EXIT
131650  105737  177564          1$:      TSTB    @#TTCSR         ;TEST FOR PRINTER READY BIT
131654  100375                           BPL     1$              ;IF NOT READY WAIT FOR IT
131656  112037  177566                   MOVB    (R0)+,@#TPB     ;WHEN READY PRINT A CHARACTER
131662  001372                           BNE     1$              ;IF LAST CHARACTER NOT NULL CONTINUE TYPING
131664  105737  177564          2$:      TSTB    @#TTCSR         ;WAIT FOR PRINTER TO FINISH
131670  100375                           BPL     2$
131672  000207                  3$:      RTS     PC

                                ;*******************************************************************
                                ;*MESSAGES
                                ;*
                                ;*******************************************************************

131674  047520  042527  020122  PWRMSG: .ASCIZ  /POWER FAILED/<12><15>
131702  040506  046111  042105
131710  006412     000
131713     105  042116  047440  ENDMSG: .ASCIZ  /END OF PASS CJKDEB0/<12><15>
131720  020106  040520  051523
131726  041440  045512  042504
131734  030102  006412     000

        000001                  .END
```

```
A        020370      5744#  5750*  5751   5754   5758   5768*  5769   5772   5776   5786*  5787*  5788*  5789
                     5792   5796   5800   5805   5813*  5814*  5815   5816*  5819   5823   5829   5833   5859
                     5861   5869*  5870   5872   5880*  5881*  5882   5890*  5891*  5892
AAADON   070110     15378  15441#
AAA1     067310     15213#
AAA10    067640     15323#
AAA11    067670     15337#
AAA12    067720     15351#
AAA13    067750     15365#
AAA2     067340     15223#
AAA3     067370     15234#
AAA4     067420     15245#
AAA5     067450     15257#
AAA6     067500     15268#
AAA7     067530     15282#
AAA8     067560     15296#
AAA9     067610     15310#
AABBF0   104106     18629  18633  18640  18646  18655#
AABDON   104126     18648  18664#
AABTP1   104116     18628  18660#
AAB2     104104     18642  18645  18647  18651#
AACDON   110750     19998  20007#
AACTP1   110742     19986  19994  20002#
AAC10    110746     19989  19995  19997  20004#
AAC2     110716     19988  19990#
AADAT0   061572     13813  13832  13854  13863#
AADBF    120066     21807* 21808  21811  21815#
AADDON   120072     21812  21818#
AADONE   061672     13862  13895#
AAD1     120012     21801#
AAPAT0   061602     13807  13827  13845  13867#
AAPAT1   061612     13810  13829  13871#
AAPAT2   061622     13815  13875#
AAPAT3   061632     13834  13879#
AAPAT4   061642     13883#
AAPAT5   061652     13848  13887#
AAPAT6   061662     13856  13891#
AA11     061462     13830#
AA12     061464     13832#
AA13     061502     13836# 13839
AA2      061412     13811#
AA20     061510     13837  13839#
AA22     061534     13849#
AA23     061536     13851#
AA24     061556     13858# 13861
AA27     061564     13859  13861#
AA3      061414     13813#
AA4      061432     13817# 13820
AA7      061440     13818  13820#
ABASE  = 000000       564
ACDW1  = 000000       564
ACDW2  = 000000       564
ACPUOP=  000000       564    579
ADC1     014334      4549   4550   4552#
ADC2     014336      4551   4554#
ADC3     014356      4558   4559   4561#
```

```
ADC4      014360     4560    4563#
ADC5      014376     4566    4567     4568     4570#
ADDW0 =   000000      564
ADDW1 =   000000      564
ADDW10=   000000      564
ADDW11=   000000      564
ADDW12=   000000      564
ADDW13=   000000      564
ADDW14=   000000      564
ADDW15=   000000      564
ADDW2 =   000000      564
ADDW3 =   000000      564
ADDW4 =   000000      564
ADDW5 =   000000      564
ADDW6 =   000000      564
ADDW7 =   000000      564
ADDW8 =   000000      564
ADDW9 =   000000      564
ADD1      014216     4504    4505     4507#
ADD2      014220     4506    4509#
ADD3      014234     4512    4513     4515#
ADD4      014236     4514    4517#
ADD5      014254     4520    4521     4523#
ADD6      014256     4522    4525#
ADD7      014270     4526    4527     4529#
ADD8      014272     4528    4531#
ADD9      014312     4534    4535     4536     4538#
ADEVCT=   000000      564     570
ADEVM =   000000      564
ADONE     050644    11581#
AENV  =   000000      564     575
AENVM =   000000      564     576
AERR1     050634    11558   11559    11560    11576#
AFATAL=   000000      564     567
AMADR1=   000000      564
AMADR2=   000000      564
AMADR3=   000000      564
AMADR4=   000000      564
AMAMS1=   000000      564
AMAMS2=   000000      564
AMAMS3=   000000      564
AMAMS4=   000000      564
AMSGAD=   000000      564     572
AMSGLG=   000000      564     573
AMSGTY=   000000      564     566
AMTYP1=   000000      564
AMTYP2=   000000      564
AMTYP3=   000000      564
AMTYP4=   000000      564
APASS =   000000      564     569
APRIOR=   000000      564
APTENV=   000001      549#   6650
AROUN     001514      726     729#
AROUND    025046     6793    6804#
ASLB1     015714     5016    5018#
ASLB2     015702     5010    5011     5013#
```

```
ASLB3   015704      5012    5015#
ASLB4   015726      5019    5020    5021    5023#
ASL1    015306      4866    4867    4868    4870#
ASL2    015310      4869    4872#
ASL3    015326      4875    4876    4877    4879#
ASL4    015330      4878    4881#
ASL5    015344      4884    4885    4887#
ASL6    015346      4886    4889#
ASL7    015372      4892    4893    4894    4895    4898#
ASRB1   015752      5036    5038#
ASRB2   015764      5039    5040    5041    5043#
ASRB3   015766      5042    5045#
ASRB4   015774      5046    5048#
ASRB5   015776      5047    5050#
ASRB6   016016      5053    5054    5055    5057#
ASRB7   016020      5056    5059#
ASR1    015416      4908    4909    4910    4912#
ASR2    015420      4911    4914#
ASR3    015442      4918    4919    4920    4922#
ASR4    015444      4921    4924#
ASR5    015460      4927    4928    4929    4931#
ASR6    015462      4930    4933#
ASR7    015512      4937    4938    4939    4940    4943#
ASTART  025536      6961#   7135    7139
ASWREG= 000000       564     577
ATESTN= 000000       564     568
ATRAP   023440      6477    6496#
AUNIT = 000000       564     571
AUSWR = 000000       564     578
AUTO1   023532      6522#
AVECT1= 000000       564
AVECT2= 000000       564
A1      050604     11564#  11580
A11     050604     11565#
A12     050620     11570#
A2      050636     11575   11579#
BBBDON  070364     15530   15540#
BBBER1  070160     15460   15473#
BBBP1   070344     15461   15463   15486   15509   15533#
BBBP2   070354     15484   15505   15535#
BBB0    070114     15456#
BBB1    070142     15465#
BBB2    070174     15477   15481#
BBB3    070214     15487#
BBB4    070246     15498   15502#
BBB5    070274     15511#  15514
BBB6    070302     15508   15514#
BBB7    070162     15472   15475#
BBCDON  111024     20033   20044#
BBCTP1  111016     20021   20029   20038#
BBC10   111022     20023   20030   20032   20042#
BBC2    110772     20025#
BBDATO  063152     14089   14109   14130   14150   14169   14188   14199#
BBDDON  120372     21885   21894#
BBDONE  063302     14197   14243#
BBD1    120152     21848#
```

```
BBPAT0   063162         14121   14132   14203#
BBPAT1   063172         14086   14106   14126   14146   14166   14185   14207#
BBPAT2   063202         14084   14091   14211#
BBPAT3   063212         14142   14215#
BBPAT4   063222         14104   14219#
BBPAT5   063232         14164   14223#
BBPAT6   063242         14183   14190   14227#
BBP10    063262         14111   14235#
BBP11    063272         14171   14239#
BBP7     063252         14152   14231#
BB10     062630         14113#  14116
BB13     062636         14114   14116#
BB14     062646         14118   14121#
BB15     062666         14127#
BB16     062712         14134#  14137
BB17     062720         14135   14137#
BB2      062526         14087#
BB20     062730         14139   14142#
BB21     062750         14147#
BB22     062774         14154#  14157
BB25     063002         14155   14157#
BB26     063012         14159   14162#
BB27     063032         14167#
BB3      062532         14089#
BB30     063054         14173#  14176
BB31     063062         14174   14176#
BB32     063072         14178   14181#
BB33     063112         14186#
BB34     063134         14192#  14195
BB35     063142         14193   14195#
BB4      062550         14093#  14096
BB5      062556         14094   14096#
BB6      062566         14099   14102#
BB7      062606         14107#
BDONE    050700         11607#
BELL   = 000240           537#
BIC1     013576          4297    4298    4300#
BIC2     013600          4299    4302#
BIC3     013616          4305    4306    4308#
BIS1     013642          4318    4319    4320    4322#
BIS2     013644          4321    4324#
BIS3     013664          4327    4328    4329    4331#
BITCHK   020044          5626#
BITCLR   017772          5612#
BITCON   020120          5639    5642#
BITSET   020010          5619#
BITST1   013532          4275    4276    4278#
BITST2   013534          4277    4281#
BITST3   013552          4284    4285    4287#
BIT0   = 000001          9280#   9851    9875    9899    9924    9945    9975    9999    10023   10047   10089   21932#  22871
                        22874   22875   22878   22879   22888   22890   23273   23292   23296
BIT1   = 000002         21933#
BIT10  = 002000         21942#
BIT11  = 004000         21943#
BIT12  = 010000         21944#
BIT13  = 020000         21945#  23297
```

```
BIT14 = 040000          21946#  22438   23248
BIT15 = 100000          21947#  22441   23251   23300
BIT2  = 000004          21934#  22074   22077   22078   22081   22082   22085   22087   22256   22263   22279   22284   22297
                        22304   22307   22326   22333   22339   22342   22357   22361   22387   22391   22411   22430   22437
                        22456   22464   22468   23434
BIT3  = 000010          21935#
BIT4  = 000020           9898    9944    9974    9998   10022   10046   10088   21936#
BIT5  = 000040          21937#
BIT6  = 000100          21862   21876   21938#  22099   22103   22104   22107   22108   22112   22114   22127   22131   22132
                        22135   22136   22140   22142   22153   22163   22168   22177   22183   22188   22202   22210   22219
                        22230   22234   22241   22295   22296   22310   22313   22330   22335   22362   22369   22386   22395
                        22410   22416   22419   22511   22515   22516   22519   22520   22528   22530   22574   22592   22600
                        22623   22637   22655   22661   22676   22682   22700   22902   22906   22907   22910   22911   22919
                        22921   22934   22938   22939   22942   22943   22951   22953   22964   22974   22979   22988   22994
                        22999   23013   23021   23030   23046   23050   23057   23113   23114   23125   23128   23144   23149
                        23172   23179   23196   23203   23223   23228   23231   23380   23383   23435   23436   23438   23439
                        23441   23472   23486   23487   23488
BIT7  = 000200          21939#  22548   22549   22551   22552   22573   22591   22622   22654   22675   22679
BIT8  = 000400          21940#
BIT9  = 001000          21941#
BRCT    001540            713*    741#    747*
BRH     001524            714*    734#
BRTAB   021162            711    5956#
BR1     021362           6043    6046#
BR10    021524           6089    6092#
BR11    021604           6107    6110#
BR12    021644           6116    6119#
BR13    021704           6125    6128#
BR14    021744           6134    6137#
BR15    022022           6151    6154#
BR16    022034           6155    6158#
BR17    022046           6159    6162#
BR2     021402           6049    6052#
BR20    022060           6163    6165#
BR21    022072           6166    6169#
BR22    022104           6170    6173#
BR23    022116           6174    6177#
BR3     021416           6055    6058#
BR4     021434           6062    6065#
BR5     021452           6069    6072#
BR6     021470           6076    6078#
BR7     021506           6082    6085#
BR70    023260           6441#
BR71    024016           6571    6578#
BTCON   020156           5658    5661#
BTERR   020150           5659#
BTRAP   023506           6504    6516#
BUF1    130454          23437   23499   23544#
BUF2    131054          23440   23510   23545#
B1      050654          11595#  11606
B2      050656          11598#
B3      050676          11604   11606#
CARRY1  003206           1084    1086#
CC    = 177776            544#   6828*
CCBDON  104166          18685   18689#
CCB10   104164          18682   18684   18687#
```

```
CCB2     104140        18677#
CCCDON   071116        15659    15719#
CCC1     070370        15555#
CCC10    070654        15627#
CCC11    070700        15635#
CCC12    070724        15644#
CCC13    070750        15652#
CCC2     070414        15563#
CCC3     070440        15571#
CCC4     070464        15578#
CCC5     070510        15586#
CCC6     070534        15594#
CCC7     070560        15602#
CCC8     070604        15611#
CCC9     070630        15619#
CCDATO   062342        13913    13932    13952    13972    13991    14010    14021#
CCERR    020342         5727#
CCP0     062352        13908    13927    13944    13986    14005    14025#
CCP1     062362        13929    14029#
CCP10    062452        13974    14057#
CCP11    062462        13993    14061#
CCP12    062472        14065#
CCP2     062372        13910    13915    14033#
CCP3     062402        13964    13988    14037#
CCP4     062412        14007    14012    14041#
CCP5     062422        13968    14045#
CCP6     062432        13948    13954    14049#
CCP7     062442        13934    14053#
CCXDON   062502        14019    14070#
CCX12    062026        13937    13939#
CCX13    062036        13941    13944#
CCX14    062056        13949#
CCX15    062102        13956#   13959
CCX18    062110        13957    13959#
CCX19    062120        13961    13964#
CCX2     061716        13911#
CCX20    062140        13969#
CCX21    062164        13976#   13979
CCX24    062172        13977    13979#
CCX25    062202        13981    13984#
CCX26    062222        13989#
CCX27    062244        13995#   13998
CCX3     061740        13917#   13920
CCX30    062252        13996    13998#
CCX31    062262        14000    14003#
CCX32    062302        14008#
CCX33    062324        14014#   14017
CCX36    062332        14015    14017#
CCX6     061746        13918    13920#
CCX7     061756        13922    13925#
CCX8     061776        13930#
CCX9     062020        13936#   13939
CC1      001520          716*     718*     732#
CC2      020252         5701*    5709#    5712*    5718*
CC3      020236         5700*    5705#    5713*    5714     5716
CDONE    051130        11692    11694#
```

```
CHECK1   130234      23496#
CHECK2   130272      23507#
CHKAPT   050426       9549   11498#  21848
CIS      025334       6816    6870#
CISADR   025040       6799#   6808    6809
CISTRP   025024       6796#   6806
CLRCD    020234       5704#   5715    5719
CLR1     014072       4435    4436    4437    4439#
CMPARE   124364      22716   22719#
CMPREG   050244       9718    9725    9732    9739   11434#
CMPSUB   070004      15213   15223   15234   15245   15257   15268   15282   15296   15310   15323   15337   15351   15365
                     15404#
CMPTMP   070100      15425   15436#
CMP1     014514       4626    4627    4629#
CMP2     014516       4628    4631#
CMP3     014540       4635    4636    4638#
CMP4     014542       4637    4640#
CMP5     014566       4644    4645    4646    4648#
CMP6     014570       4647    4650#
CMP7     014610       4653    4654    4656#
COMHLT   130452        683*  23542#
COM1     014632       4667    4668    4670#
CONCIS   025100       6797    6812#
CONT     001542        724     728     742#
CONT1    123520      22556#  22559
CONT42   127064      23276   23281#
CON1     020260       5710    5712#
CON2     020344       5726    5729#
CORH     023500       6497*   6510#
COUNT    025414       6901#   6902    6938*   6973*   6978    7002*   7007    7030*   7035    7058*   7063    7087*   7093
                      7117*   7122    7388*   7410*   7420    7447*   7456    7484*   7493
CPUHLT   002130        677*    816     832#
CTRAP    023456       6502#   6521
C15      050710      11621#
C2       050724      11626   11628#
C25      050732      11631#
C3       050750      11636   11638#
C35      050756      11641#
C4       050774      11646   11648#
C45      051000      11650#
C5       051016      11655   11657#
C55      051022      11660#
C6       051036      11665   11667#
C65      051044      11669#
C7       051062      11674   11676#
C75      051070      11678#
C8       051106      11683   11685#
C85      051112      11687#
DADTBL   127304      23346#  23374   23375   23381
DAERR    020116       5627    5629    5631    5633    5635    5637    5640#
DALTB1   044126      10439   10495#
DALTB2   044162      10453   10513#
DALTB3   044412      10533   10589#
DALTB4   044446      10547   10607#
DBE      024646       6745#   6749
DBE1     024654       6740    6749#
```

DBE2	024652	6742	6747#						
DBE3	024666	6750	6752#						
DBE4	024700	6753	6755#						
DBE5	024710	6756	6758#						
DDBBFO	104352	18744	18767#						
DDBDON	104362	18753	18772#						
DDBTP1	104332	18730	18758#						
DDBTP2	104342	18734	18762#						
DDB2	104266	18739#							
DDB5	104330	18749	18752	18754#					
DDCDON	111124	20068	20077#						
DDCTP1	111110	20058	20064	20072#					
DDC10	111122	20061	20065	20067	20075#				
DDC2	111056	20060	20062#						
DDDATO	064052	14261	14281	14301	14321	14340	14359	14379	14391#
DDDDON	071610	15811	15874#						
DDDONE	064202	14389	14435#						
DDD1	071122	15734#							
DDD2	071162	15745#							
DDD3	071222	15756#							
DDD4	071262	15767#							
DDD5	071322	15778#							
DDD6	071362	15790#							
DDD7	071422	15801#							
DDONE	051276	11729	11752#						
DDP0	064062	14283	14303	14395#					
DDP1	064072	14256	14258	14278	14296	14399#			
DDP2	064102	14276	14298	14403#					
DDP3	064112	14316	14337	14407#					
DDP4	064122	14354	14376	14411#					
DDP5	064132	14356	14374	14415#					
DDP6	064142	14318	14335	14419#					
DDP7	064152	14323	14342	14423#					
DDP8	064162	14361	14381	14427#					
DDP9	064172	14263	14431#						
DD10	063434	14285#	14288						
DD11	063442	14286	14288#						
DD12	063456	14291	14294#						
DD13	063476	14299#							
DD14	063520	14305#	14308						
DD15	063526	14306	14308#						
DD16	063542	14311	14314#						
DD17	063562	14319#							
DD18	063604	14325#	14328						
DD2	063326	14259#							
DD21	063612	14326	14328#						
DD22	063622	14330	14333#						
DD23	063642	14338#							
DD24	063664	14344#	14347						
DD27	063672	14345	14347#						
DD3	063350	14265#	14268						
DD30	063702	14349	14352#						
DD31	063722	14357#							
DD32	063744	14363#	14366						
DD35	063752	14364	14366#						
DD36	063766	14369	14372#						

DD37	064006	14377#												
DD38	064030	14383#	14386											
DD41	064036	14384	14386#											
DD6	063356	14266	14268#											
DD7	063372	14271	14274#											
DD8	063412	14279#												
DEC1	013772	4385	4386	4387	4389#									
DEC2	013774	4388	4391#											
DEC3	014010	4394	4395	4397#										
DEC4	014012	4396	4399#											
DEC5	014026	4402	4403	4405#										
DEC6	014030	4404	4407#											
DEC7	014052	4411	4412	4413	4415#									
DERR1	051232	11708	11733#											
DERR2	051200	11707	11720#											
DISPRE	000174	638#												
DIVDSU	071466	15734	15745	15756	15767	15778	15790	15801	15840#					
DIVDT	071600	15856	15860	15871#										
DIVFSU	071000	15555	15563	15571	15578	15586	15594	15602	15611	15619	15627	15635	15644	15652
		15687#												
DIVFT	071106	15703	15716#											
DL11W	023600	6532	6535#											
DNMB0A	007412	2941	2942	2943	2945#									
DNMB0B	007414	2944	2947#											
DNMB2A	007546	3000	3001	3002	3004#									
DNMB2B	007550	3003	3006#											
DNMB2C	007556	3007	3009#											
DNMB2D	007572	3011	3012	3014#										
DNMB2E	007574	3013	3016#											
DNMB2F	007604	3018	3020#											
DNMB3A	007650	3037	3038	3039	3041#									
DNMB3B	007652	3040	3043#											
DNMB3C	007662	3044	3046#											
DNMB3D	007700	3049	3050	3052#										
DNMB3E	007702	3051	3054#											
DNMB4A	010022	3092	3093	3094	3096#									
DNMB4B	010024	3095	3098#											
DNMB4C	010034	3099	3101#											
DNMB4D	010044	3102	3104#											
DNMB4E	010046	3103	3106#											
DNMB4F	010054	3107	3109#											
DNM03A	007012	2688	2689	2690	2692#									
DNM03B	007014	2691	2694#											
DNM03C	007024	2695	2698#											
DNM1	006724	2655	2657#											
DNM1A	007444	2959	2960	2961	2963#									
DNM1B	007446	2962	2965#											
DNM2	006732	2658	2660#											
DNM2A	007476	2977	2978	2980#										
DNM2B	007500	2979	2982#											
DNM2C	007506	2985#												
DNM2D	007510	2984	2987#											
DNM3	006742	2662	2664#											
DNM4	006756	2668	2670#											
DNM4A	007746	3069	3070	3071	3073#									
DNM4B	007750	3072	3075#											

DNM4C	007756	3076	3078#							
DNM5A	010116	3123	3124	3125	3127#					
DNM5B	010120	3126	3129#							
DNM5C	010130	3130	3132#							
DNM6A	010172	3145	3146	3147	3149#					
DNM6B	010174	3148	3151#							
DNM6C	010204	3152	3154#							
DNM7A	010250	3167	3168	3169	3171#					
DNM7B	010252	3170	3173#							
DNM7C	010262	3174	3176#							
DOAGIN	131532	23559	23565#							
DOPB2A	007232	2851	2853#							
DOPB2B	007264	2874	2876#							
DOP0A	006560	2587	2589#							
DOP0B	006576	2594	2596#							
DOP0C	006610	2599	2601#							
DOP0D	006632	2608	2610#							
DOP03A	006672	2631	2632	2633	2635#					
DOP03B	006674	2634	2637#							
DOP1	007132	2782	2784#							
DOP2	007202	2826	2828#							
DOP4	011244	3491#								
DOP5	011312	3520#								
DPAT3	056024	12922#								
DTRAP1	023476	6508#	6520							
DUMMY =	000000	6897#	7613	7633	7654	7675	7695	7716	7737	
D1	051154	11710#	11725	11731						
D2	051166	11713*	11714#							
D3	051170	11715#	11733							
D4	051174	11717#								
D5	051202	11722#	11748							
D6	051216	11723	11727#							
D7	051226	11728	11730#							
D8	051262	11742#								
D9	051274	11747	11750#							
EDONE	051344	11776	11779#							
EEBBF0	104476	18813#								
EEBBF1	104506	18785	18791	18796	18802	18817#				
EEBDON	104520	18806	18823#							
EEBTP1	104466	18784	18809#							
EEB10	104516	18792	18799	18803	18805	18821#				
EEB2	104426	18793#								
EECDON	111232	20097	20110#							
EECTP1	111206	20088	20089*	20103#	20105					
EECTP2	111216	20087	20093	20105#						
EEC10	111230	20090	20094	20096	20108#					
EEC2	111154	20091#								
EEDATO	064356	14453	14473	14485#						
EEDONE	064440	14483	14510#							
EEEDON	072342	15992	16052#							
EEE1	071614	15889#								
EEE10	072100	15961#								
EEE11	072124	15969#								
EEE12	072150	15977#								
EEE13	072174	15985#								
EEE2	071640	15897#								

```
EEE3      071664          15905#
EEE4      071710          15913#
EEE5      071734          15921#
EEE6      071760          15929#
EEE7      072004          15937#
EEE8      072030          15945#
EEE9      072054          15953#
EEP0      064366          14455    14475    14489#
EEP1      064400          14448    14450    14494#
EEP2      064410          14498#
EEP3      064420          14468    14470    14502#
EEP4      064430          14506#
EERR0     051342          11763    11772    11777#
EE12      064342          14478    14480#
EE2       064226          14451#
EE3       064250          14457#   14460
EE6       064256          14458    14460#
EE7       064272          14463    14466#
EE8       064312          14471#
EE9       064334          14477#   14480
EISEND    037342           9214     9216#
EMTA    = 104377            547#     885
ENDMSG    131713          23556    23611#
ENDPAS    131454          23359    23411    23523    23550#
ENT176    031224           7595     7602#
ENT51     027150           7194     7198#
ER        001536            725      727      739#
ERROR1    002062            740      768      770      772      774      776      782      784      786      797      799      801      803
                            805      811      813      815      821#     843      845      847      849      851      857      859      861
                            872      874      876      878      880      886      888      890      896     1004     6240     6250     6261
                           6262     6272     6282     6292     6305
ERROR2    002400            891      896#     909      911      913      915      917      923      925      927      944      946      948
                            950      952      958      960      962      973      975      977      979      981      987      989      991
ERROR3    050452           9310    11511#
ERROR4    120424          11549    21905#   21922
ERROR5    123150          21967    22471#
ERROR6    124412          22489    22731#
ERROR7    127234          22759    23332#
ERROR8    130404          23361    23531#
ERRVEC= 000004             9281#   10258*   10271*   11560*   11707*   11803*   11836*   12717*   13017*   13079*   13145*   15063*   17934*
                          17996*   18057*   18110*   18157*   18203*   18792*   18843*   18890*   18941*   18992*   19044*   19090*   19139*
                          19183*   19989*   20023*   20061*   20090*   20123*   20155*   20188*   20222*   21016*   21055*   21094*   21132*
                          21173*   21213*   21253*   21923*
E1        051310          11765#
E3        051316          11767#
E4        051320          11768#
F       = 000063           6898#    7206     7223#    7240#    7257#    7274#    7291#    7308#    7325#    7342#    7359#    7376#
FDATI0    051620          11795    11817    11864#
FDATI1    051622          11865#
FDATI2    051624          11866#
FDATI3    051626          11867#
FDATI4    051630          11802    11811    11868#
FDATI5    051632          11869#
FDATI6    051634          11870#
FDATI7    051636          11871#
FDATO0    051642          11831    11850    11873#
```

```
FDATO1  051644    11874#
FDATO2  051646    11875#
FDATO3  051650    11876#
FDATO4  051652    11835   11844   11877#
FDATO5  051654    11878#
FDATO6  051656    11879#
FDATO7  051660    11880#
FDONE   051704    11860   11892#
FERR20  051616    11803   11836   11861#
FFBBF1  104634    18836   18842   18848   18854   18865#
FFBDON  104646    18858   18871#
FFBTP1  104624    18835   18861#
FFB10   104644    18843   18851   18855   18857   18869#
FFB2    104564    18845#
FFCDON  111332    20130   20142#
FFCTP1  111310    20121   20122*  20135#  20137
FFCTP2  111320    20120   20126   20137#
FFC10   111330    20123   20127   20129   20140#
FFC2    111264    20124#
FFDATO  064604    14528   14548   14561#
FFDONE  064664    14557   14587#
FFFDON  072674    16110   16173#
FFF1    072346    16067#
FFF2    072406    16078#
FFF3    072446    16089#
FFF4    072506    16100#
FFP0    064614    14543   14566#
FFP1    064624    14545   14570#
FFP2    064634    14523   14574#
FFP3    064644    14525   14578#
FFP4    064654    14530   14550   14582#
FF10    064574    14553   14555#
FF2     064464    14526#
FF3     064506    14532#  14535
FF4     064514    14533   14535#
FF5     064524    14537   14541#
FF6     064544    14546#
FF7     064566    14552#  14555
FINIE   130330    23518#
FINISH  025400     6792*   6822    6888#
FOVER   022700     6314    6326#
FPEXIT  120420    21849   21902#
FPHLT   120472      679*  21916#
FPP     025370     6792    6820    6884#
FPSTRT  050522     9309   11357   11545#
FPVECT= 000244    11533#  11558*  11708*  11763*  12712*  12742*  13322*  13400*  14784*  14806*  14842*  14864*  14898*
                  14920*  15460*  15508*  16274*  16419*  16549*  16700*  17737*  21922*
FXDATO  051664    11796   11818   11851   11882#
FXDAT1  051666    11883#
FXDAT2  051670    11884#
FXDAT3  051672    11885#
FXDAT4  051674    11886#
FXDAT5  051676    11887#
FXDAT6  051700    11888#
FXDAT7  051702    11889#
F10     051532    11840#
```

```
F11      051534      11841#
F12      051550      11845   11847#
F13      051574      11853#  11856
F135     051560      11848   11850#
F14      051602      11854   11856#
F2       051372      11799#  11800
F22      051604      11857#
F3       051412      11807#
F4       051414      11808#
F5       051430      11812   11814#
F6       051476      11827   11829#
F7       051512      11832#  11833
GAND0    054272      12482#
GAND1    054274      12483#
GAND2    054276      12484#
GAND3    054300      12485#
GCMP     054222      12124   12150   12176   12202   12228   12254   12280   12306   12332   12358   12384   12410   12457#
GDAT00   054312      12116   12142   12168   12194   12220   12246   12272   12298   12324   12350   12376   12402   12444
                     12450   12457   12492#
GDAT01   054314      12493#
GDAT02   054316      12494#
GDAT03   054320      12495#
GDONE    054322      12427   12498#
GFLAG1   054246      12126   12128*  12152   12154*  12178   12180*  12204   12206*  12230   12232*  12256   12258*  12282
                     12284*  12308   12310*  12334   12336*  12360   12362*  12386   12388*  12412   12414*  12430   12469#
GFLAG2   054250      12470#
GGBBF0   104762      18883   18895   18901   18912#
GGBBF1   104772      18889   18916#
GGBDON   105004      18905   18922#
GGBTP1   104752      18882   18908#
GGB10    105002      18890   18898   18902   18904   18920#
GGB2     104712      18892#
GGCDON   111426      20164   20174#
GGCTP1   111412      20152   20153*  20160   20169#
GGC10    111424      20155   20159   20161   20163   20172#
GGC2     111360      20156#
GGDAT0   066322      14790   14820   14848   14876   14904   14934   14952#
GGDONE   066442      14950   14994#
GGER0    065570      14784   14812#  14842
GGER14   066226      14898   14926#
GGGDON   075460      16879   16964#
GGG1     074530      16755#
GGG10    075060      16836#
GGG11    075110      16845#
GGG12    075140      16854#
GGG13    075170      16863#
GGG14    075220      16872#
GGG2     074560      16764#
GGG3     074610      16773#
GGG4     074640      16782#
GGG5     074670      16791#
GGG6     074720      16800#
GGG7     074750      16809#
GGG8     075000      16818#
GGG9     075030      16827#
GGP1     066332      14957#
```

GGP2	066342	14843	14865	14899	14921	14961#								
GGP3	066352	14845	14867	14965#										
GGP4	066362	14969#												
GGP5	066372	14785	14787	14807	14809	14973#								
GGP6	066402	14792	14822	14850	14906	14977#								
GGP7	066412	14878	14982#											
GGP8	066422	14901	14923	14986#										
GGP9	066432	14936	14990#											
GG10	065664	14836	14840#											
GG11	065712	14846#												
GG12	065734	14852#	14855											
GG13	065742	14853	14855#											
GG14	065756	14858	14862#											
GG15	066004	14868#	14870											
GG16	066010	14864	14870#											
GG17	066044	14880#	14883											
GG18	066052	14881	14883#											
GG19	066102	14892	14896#											
GG2	065472	14788#												
GG20	066130	14902#												
GG21	066152	14908#	14911											
GG22	066160	14909	14911#											
GG23	066174	14914	14918#											
GG24	066222	14924#	14928											
GG25	066230	14920	14928#											
GG26	066264	14938#	14941											
GG27	066272	14939	14941#											
GG3	065514	14794#	14797											
GG4	065522	14795	14797#											
GG5	065536	14800	14804#											
GG6	065564	14810#	14814											
GG7	065572	14806	14814#											
GG8	065626	14824#	14827											
GG9	065634	14825	14827#											
GIN1	025112	6814#	6821	6845										
GIN2	025154	6825#	6846											
GIN3	025202	6823	6830#											
GOR0	054302	12487#												
GOR1	054304	12488#												
GOR2	054306	12489#												
GOR3	054310	12490#												
GPAT00	054252	12115	12167	12219	12271	12323	12375	12440	12472#					
GPAT01	054254	12473#												
GPAT02	054256	12474#												
GPAT03	054260	12475#												
GPAT10	054262	12141	12193	12245	12297	12349	12401	12435	12477#					
GPAT11	054264	12478#												
GPAT12	054266	12479#												
GPAT13	054270	12480#												
GRESET	054202	12136	12162	12188	12214	12240	12266	12292	12318	12344	12370	12396	12422	12450#
GSETUP	054124	12117	12143	12169	12195	12221	12247	12273	12299	12325	12351	12377	12403	12430#
GS1	054154	12440#	12454											
G1	052540	12119#	12138											
G10	052772	12179	12183#											
G11	052774	12182	12184#											
G12	053040	12197#	12216											

```
G13        053072          12205    12209#
G14        053074          12208    12210#
G15        053140          12223#   12242
G16        053172          12231    12235#
G17        053174          12234    12236#
G2         052572          12127    12131#
G20        053240          12249#   12268
G21        053272          12257    12261#
G22        053274          12260    12262#
G23        053340          12275#   12294
G24        053372          12283    12287#
G25        053374          12286    12288#
G26        053440          12301#   12320
G27        053472          12309    12313#
G3         052574          12130    12132#
G30        053474          12312    12314#
G31        053540          12327#   12346
G32        053572          12335    12339#
G33        053574          12338    12340#
G34        053640          12353#   12372
G35        053672          12361    12365#
G36        053674          12364    12366#
G37        053740          12379#   12398
G4         052640          12145#   12164
G40        053772          12387    12391#
G41        053774          12390    12392#
G42        054040          12405#   12424
G43        054072          12413    12417#
G44        054074          12416    12418#
G5         052672          12153    12157#
G6         052674          12156    12158#
G7         052740          12171#   12190
HADR       055014          12643*   12651    12660#
HA1R       055070          12621    12645    12654    12674#
HA1W       055020          12511    12521    12548    12644    12663#
HA2R       055100          12625    12676#
HA2W       055030          12525    12561    12665#
HA3R       055110          12629    12678#
HA3W       055040          12529    12574    12667#
HA4R       055120          12633    12680#
HA4W       055050          12533    12587    12669#
HA5R       055130          12637    12682#
HA5W       055060          12537    12600    12671#
HCLR       054776          12517    12619    12654#
HCLR1      055006          12656#   12657
HCMP       054742          12543    12555    12568    12581    12594    12607    12643#
HCMP1      054762          12647#   12650
HCMP2      054770          12648    12650#
HDAT1      055140          12512    12685#
HDAT2      055150          12687#
HDAT3      055160          12689#
HDAT4      055170          12691#
HDAT5      055200          12693#
HDONE      055210          12613    12696#
HERE     = 000000           6018#    6041*    6047*    6093
HFLAG      055016          12510*   12611    12615*   12661#
```

HHBBF0	105130	18934	18946	18961	18963#		
HHBBF1	105140	18940	18951	18967#			
HHBDON	105152	18955	18973#				
HHBTP1	105110	18933	18958#				
HHB10	105150	18941	18949	18952	18954	18971#	
HHB2	105050	18943#					
HHCDON	111536	20197	20208#				
HHCTP1	111514	20185	20186*	20202#			
HHCTP2	111524	20184	20193	20204#			
HHC10	111534	20188	20192	20194	20196	20206#	
HHC2	111462	20189#					
HHDATO	065270	14617	14641	14664	14684	14704	14716#
HHDONE	065440	14714	14768#				
HHHDON	076674	17147	17230#				
HHH1	075464	16980#					
HHH10	076234	17097#					
HHH11	076304	17110#					
HHH12	076354	17123#					
HHH13	076424	17136#					
HHH2	075534	16993#					
HHH3	075604	17006#					
HHH4	075654	17019#					
HHH5	075724	17032#					
HHH6	075774	17045#					
HHH7	076044	17058#					
HHH8	076114	17071#					
HHH9	076164	17084#					
HHP0	065300	14612	14720#				
HHP1	065310	14614	14724#				
HHP10	065420	14686	14760#				
HHP11	065430	14706	14764#				
HHP2	065320	14619	14728#				
HHP3	065330	14732#					
HHP4	065340	14643	14736#				
HHP5	065350	14636	14699	14701	14740#		
HHP6	065360	14638	14744#				
HHP7	065370	14666	14748#				
HHP8	065400	14659	14679	14681	14752#		
HHP9	065410	14661	14756#				
HH10	065020	14646	14648#				
HH11	065034	14651	14656#				
HH12	065054	14662#					
HH13	065076	14668#	14671				
HH16	065104	14669	14671#				
HH17	065120	14674	14677#				
HH18	065140	14682#					
HH19	065162	14688#	14691				
HH2	064710	14615#					
HH20	065170	14689	14691#				
HH21	065204	14694	14697#				
HH22	065224	14702#					
HH23	065246	14708#	14711				
HH24	065254	14709	14711#				
HH3	064732	14621#	14624				
HH6	064740	14622	14624#				
HH7	064750	14626	14633#				

```
HH8        064770     14639#
HH9        065012     14645#   14648
HICORE     023372      6481#    6499
HLT      = 000000       525#
HSTD       054664     12541    12554    12567    12580    12593    12606    12619#
HXDATO     067174     15018    15046    15080    15097    15121    15143    15157#
HXDONE     067304     15154    15197#
HXER9      066700     15063    15071#
HXP1       067204     15009    15034    15112    15116    15124    15140    15145    15162#
HXP2       067214     15011    15013    15037    15040    15077    15167#
HXP3       067224     15171#
HXP4       067234     15094    15175#
HXP5       067244     15100    15179#
HXP6       067254     15075    15092    15138    15183#
HXP7       067264     15020    15082    15188#
HXP8       067274     15049    15192#
HX10       066576     15041    15044#
HX11       066620     15051#   15054
HX14       066626     15052    15054#
HX15       066642     15058    15061#
HX16       066660     15065#
HX165      066664     15066#
HX17       066702     15070    15073#
HX18       066722     15078#
HX19       066742     15084#   15087
HX2        066466     15012#
HX20       066750     15085    15087#
HX22       066772     15095#
HX23       067012     15102#   15105
HX26       067020     15103    15105#
HX28       067044     15117#
HX29       067066     15126#   15129
HX3        066500     15014    15016#
HX30       067074     15127    15129#
HX31       067110     15133    15136#
HX32       067130     15141#
HX33       067152     15147#   15150
HX34       067160     15148    15150#
HX4        066520     15022#   15025
HX7        066526     15023    15025#
HX8        066542     15028    15031#
HX9        066564     15039#
H1         054326     12510#
H10        054570     12590#   12595
H11        054622     12603#   12608
H12        054654     12612    12615#
H2         054346     12514#   12515
H3         054356     12519#   12616
H4         054430     12541#
H5         054452     12551#   12556
H6         054504     12564#   12569
H7         054536     12577#   12582
IDATI0     052176     11906    11912    11927    11937    11942    11968    11991#
IDATI1     052200     11992#
IDATI2     052202     11965*   11993#
IDATI3     052204     11966*   11994#
```

```
IDAT00   052166     11923   11926   11962   11967   11986#
IDAT01   052170     11987#
IDAT02   052172     11988#
IDAT03   052174     11989#
IDONE    052206     11974   11996#
IIBBF0   105276     18985   18997   19012   19014#
IIBBF1   105306     18991   19002   19018#
IIBDON   105320     19006   19025#
IIBTP1   105256     18984   19009#
IIB10    105316     18992   19000   19003   19005   19023#
IIB2     105216     18994#
IICDON   111602     20234   20238#
IIC2     111560     20227#
IIC20    111600     20222   20235#
IIIDON   073176     16224   16305#
III1     072700     16188#
III2     072730     16198#
III3     072760     16207#
III4     073010     16216#
IJMP     013014      4083    4086#
IJMP4    012632      4044    4047#   4055
IJMP5    012764      4075    4078#
ILLA   = 004700       542#
ILLB   = 000100       543#
INC1     013706      4353    4354    4356#
INC2     013710      4355    4358#
INC3     013732      4362    4363    4364    4366#
INC4     013734      4365    4369#
INC5     013750      4372    4373    4375#
INST     025402      6824*   6825*   6829    6837    6844    6889#
IOHAND   130160     23449   23456   23465   23473   23483#
IPAT10   052146     11907   11936   11976#
IPAT11   052150     11977#
IPAT12   052152     11978#
IPAT13   052154     11979#
IPAT20   052156     11915   11945   11981#
IPAT21   052160     11982#
IPAT22   052162     11983#
IPAT23   052164     11984#
ITRAP5=  000004       527#
I1       051710     11904#
I105     052010     11930   11932#
I12      052012     11936#
I13      052026     11939#  11940
I14      052054     11951#
I15      052056     11952#
I16      052060     11953#
I17      052074     11957   11959#
I2       051726     11909#  11910
I20      052134     11970#  11973
I23      052142     11971   11973#
I3       051752     11919#
I4       051754     11920#
I5       051756     11921#
I6       052002     11929#  11932
JBUF0    055622     12825   12841#
```

```
JBUF1    055624      12842#
JBUF2    055626      12843#
JBUF3    055630      12844#
JDATI0   055632      12815    12828    12836    12846#
JDATI1   055634      12847#
JDATI2   055636      12848#
JDATI3   055640      12849#
JDATO0   055642      12822    12829    12851#
JDATO    055652      12812    12856#
JDATO1   055644      12852#
JDATO2   055646      12853#
JDATO3   055650      12854#
JDAT1    055654      12857#
JDAT2    055656      12858#
JDAT3    055660      12859#
JDONE    055662      12839    12862#
JJBBFO   105440      19037    19043    19049    19054    19065#
JJBDON   105452      19058    19071#
JJBTP1   105426      19036    19061#
JJB10    105450      19044    19052    19055    19057    19069#
JJB2     105364      19046#
JJJDON   073560      16371    16450#
JJJ1     073202      16321#
JJJ2     073246      16334#
JJJ3     073312      16347#
JJJ4     073356      16360#
JMPCK    013016       4086     4088#
JMPERR   020202       5676     5677     5678     5680#
JMPSEQ   013030       4035*    4041     4045*    4049     4052*    4058     4062*    4065     4069*    4072     4076*    4080     4084*
                      4088     4091#
JMPT     020172       5675     5676#
JMP2     012634       4036     4049#
JMP2A    012644       4050     4052#
JMP3     012576       4038#    4053
JMP3A    012606       4039     4041#
JMP3B    012620       4042     4044#
JMP4     012656       4047     4055#
JMP4A    012666       4056     4058#
JMP4B    012700       4059     4061#
JMP5     012740       4061     4072#
JMP5A    012752       4073     4075#
JMP6     012712       4065#    4078
JMP6A    012724       4066     4068#
JMP7     012766       4068     4080#
JMP7A    013000       4081     4083#
JSRCK    013400       4200     4203#
JSRCKA   013374       4197     4200#
JSRCK1   013416       4114     4204     4207#
JSRSEQ   013376       4118*    4127     4139*    4142     4154*    4158     4164*    4168     4176*    4178     4186*    4190     4196*
                      4201#    4203
JSR0     013034       4114#
JSR1     013040       4113     4116#
JSR1A    013062       4124#    4144
JSR2     013136       4117     4142#    4160
JSR2A    013166       4143     4145     4147     4150#
JSR2B    013170       4149     4152#
```

```
JSR3     013064        4127#  4135   4155
JSR3A    013124        4128   4130   4132   4134   4137#
JSR3B    013126        4136   4139#
JSR4     013212        4129   4140   4158#
JSR4A    013230        4159   4162#
JSR4B    013232        4161   4164#
JSR5     013302        4165   4178#  4182   4192
JSR5A    013326        4179   4181   4184#
JSR5B    013330        4183   4186#
JSR6     013244        4168#  4180   4199
JSR6A    013270        4169   4171   4174#
JSR6AD   013372        4172   4187   4199#  4205
JSR6B    013272        4173   4176#
JSR7     013342        4170   4177   4190#
JSR7A    013360        4191   4194#
JSR7B    013362        4193   4196#
J1       055522        12809#
J2       055540        12818#
J3       055542        12819#
J4       055544        12820#
J5       055600        12831# 12834
J6       055606        12832  12834#
J7       055620        12837  12839#
KBUF0    055776        12879  12888  12899  12909#
KBUF1    056000        12910#
KBUF2    056002        12911#
KBUF3    056004        12912#
KDATI0   055766        12891  12904#
KDATI1   055770        12905#
KDATI2   055772        12906#
KDATI3   055774        12907#
KDATO0   056006        12885  12892  12914#
KDATO1   056010        12915#
KDATO2   056012        12916#
KDATO3   056014        12917#
KDONE    056026        12902  12924#
KERSTK=  001000         545#  9386   9390   9716   9723   9730   9737   10260  10443  10464  10537  10558  10640
                       10693  10698  10751  10768  10811  10857  10939  11206  11307  11336  11355
KFIRST   124406        22717* 22719  22727#
KIPAR0=  172340         9266# 9428   9568   9640   9722   9767   9831   10905* 11411  11448
KIPAR1=  172342         9267# 10906*
KIPAR2=  172344         9268# 10907*
KIPAR3=  172346         9269# 10146  10292* 10628* 10908*
KIPAR4=  172350         9270# 9848*  9872*  9896*  9921*  9943*  9972*  9996*  10020* 10044* 10073* 10119* 10126*
                       10293* 10629* 10909* 10928* 11044* 11192*
KIPAR5=  172352         9271# 10074* 10120* 10127* 10671*
KIPAR6=  172354         9272# 10270*
KIPAR7=  172356         9273# 10910*
KIPDR0=  172300         9255# 9440   9602   9677   9714   9756   9838   10150  10153  10166  10914  11407  11435
KIPDR1=  172302         9256#
KIPDR2=  172304         9257#
KIPDR3=  172306         9258# 10296* 10437* 10630*
KIPDR4=  172310         9259# 10117* 10124* 10303* 10362* 10446* 10457* 10540* 10551* 10631* 10656* 10672* 10739*
                       10927* 10933* 11042* 11067* 11069* 11083* 11085* 11097* 11099* 11114* 11116* 11131* 11133*
                       11147* 11149* 11165* 11167* 11191*
KIPDR5=  172312         9260# 10118* 10125* 10438* 10673* 10740*
```

```
KIPDR6= 172314          9261#  10257*  10261   10269*
KIPDR7= 172316          9262#  10247*  10250   10802*  10831*
KKBBF0  105600         19083   19095   19110   19112#
KKBBF1  105610         19089   19100   19116#
KKBDON  105622         19104   19122#
KKBTP1  105560         19082   19107#
KKB10   105620         19090   19098   19101   19103   19120#
KKB2    105516         19092#
KKCDON  112700         20457   20521#
KKC1    111704         20293#
KKC10   112136         20354#
KKC11   112164         20362#
KKC12   112212         20370#
KKC13   112240         20378#
KKC14   112266         20386#
KKC15   112314         20394#
KKC16   112342         20402#
KKC17   112370         20410#
KKC2    111732         20303#
KKC20   112416         20418#
KKC21   112444         20426#
KKC22   112472         20434#
KKC23   112520         20442#
KKC24   112546         20450#
KKC3    111760         20313#
KKC4    112006         20322#
KKC5    112034         20330#
KKC6    112062         20338#
KKC7    112110         20346#
KKKDON  074112         16504   16588#
KKK1    073564         16466#
KKK3    073614         16476#
KKK4    073644         16486#
KKK5    073674         16496#
KPAT0   056016         12876   12919#
KPAT1   056020         12920#
KPAT2   056022         12921#
KWSTRT  123220         21965   22005   22469   22485#
KX2     055704         12881#
KX3     055706         12882#
KX4     055710         12883#
KX5     055744         12894#  12897
KX6     055752         12895   12897#
KX7     055764         12900   12902#
K1      021300          6020#   6102*   6103    6106    6111*   6112    6115    6119*   6121    6128*   6130    6137*   6139
                        6142
K10     021316          6027#   6158    6162    6165    6169    6173    6177
K11     021320          6028#
K12     021322          6029#
K2      021302          6021#
K3      021304          6022#
K4      021306          6023#
K5      021310          6024#   6101*   6104    6110*   6120*   6122    6124    6129*   6131    6133    6138*   6140
K6      021312          6025#   6113    6162
K7      021314          6026#   6150    6154    6158    6165    6177
LDATI0  056202         12942   12949   12967   12987#
```

```
LDATI1   056204        12988#
LDATI2   056206        12962    12965*   12989#
LDATI3   056210        12966*   12990#
LDATO0   056214        12959    12968    12992#
LDATO1   056216        12993#
LDATO2   056220        12994#
LDATO3   056222        12995#
LDCDSU   113362        20534    20544    20554    20564    20574    20584    20594    20604    20615    20625    20667#
LDCFSU   112576        20293    20303    20313    20322    20330    20338    20346    20354    20362    20370    20378    20386    20394
                       20402    20410    20418    20426    20434    20442    20450    20490#
LDCT     112670        20498    20503    20518#   20674    20680
LDONE    056224        12975    12997#
LDXSUB   114622        20708    20721    20734    20747    20761    20774    20788    20801    20814    20827    20840    20853    20867
                       20881    20895    20909    20956#
LDXT     114750        20972    20974    20993#
LKS      120552        21959#   22498    22511    22515*   22516    22519*   22520    22528*   22530    22545    22548*   22549    22551*
                       22552    22556    22573*   22574*   22575    22581*   22584    22591*   22592*   22593    22600*   22622*   22623*
                       22624    22637*   22654*   22655*   22656    22661*   22675*   22676*   22677    22679*   22682*   22700*   22705*
                       22706    22708*   22709    22712*   23441*   23488*
LKSTST   123250        22491#
LLBBFO   105726        19133    19141*   19144    19158#
LLBDON   105740        19151    19165#
LLBTP1   105716        19132    19154#
LLB10    105736        19139    19147    19150    19163#
LLB2     105662        19141#
LLCDON   113454        20634    20694#
LLC1     112704        20534#
LLC10    113226        20604#
LLC11    113264        20615#
LLC12    113322        20625#
LLC2     112742        20544#
LLC3     113000        20554#
LLC4     113036        20564#
LLC5     113074        20574#
LLC6     113132        20584#
LLC7     113170        20594#
LLLDON   074524        16654    16738#
LLL1     074116        16604#
LLL2     074162        16617#
LLL3     074226        16630#
LLL4     074272        16643#
LPAT10   056160        12939    12977#
LPAT11   056162        12978#
LPAT12   056164        12979#
LPAT13   056166        12980#
LPAT20   056170        12943    12982#
LPAT21   056172        12983#
LPAT22   056174        12984#
LPAT23   056176        12985#
LTCHLT   124460          680*   22742#
L1       056032        12936#
L2       056072        12953#
L3       056074        12954#
L4       056076        12956#
L5       056146        12971#   12974
L6       056154        12972    12974#
```

MBDM2A	010404	3245	3246	3248#	
MBDM2B	010406	3247	3250#		
MBDM2C	010416	3251	3253#		
MBDM2D	010430	3254	3255	3257#	
MBDM2E	010432	3256	3259#		
MBDM2F	010442	3260	3262#		
MBDM4A	010652	3343	3345#		
MBDM4B	010662	3346	3348#		
MBDM4C	010674	3349	3350	3352#	
MBDM4D	010676	3351	3354#		
MBDM4E	010704	3355	3357#		
MDAT00	056352	13029	13032	13053#	
MDAT01	056354	13054#			
MDAT02	056356	13055#			
MDAT03	056360	13056#			
MDM1A	010314	3195	3196	3198#	
MDM1B	010316	3197	3200#		
MDM2A	010344	3219	3220	3222#	
MDM2B	010346	3221	3224#		
MDM2C	010354	3227#			
MDM2D	010356	3226	3229#		
MDM3A	010500	3279	3280	3282#	
MDM3B	010502	3281	3284#		
MDM3C	010512	3285	3287#		
MDM3D	010524	3288	3290#		
MDM3E	010544	3292	3294#		
MDM4A	010606	3314	3315	3317#	
MDM4B	010610	3316	3319#		
MDM4C	010616	3320	3322#		
MDM5A	010736	3378	3379	3381#	
MDM5B	010740	3380	3383#		
MDM5C	010750	3384	3386#		
MDM5D	010760	3387	3389#		
MDM5E	011000	3392	3394#		
MDM6A	011032	3414	3415	3417#	
MDM6B	011034	3416	3419#		
MDM6C	011044	3420	3422#		
MDM6D	011056	3423	3425#		
MDM6E	011100	3428	3430#		
MDM7A	011136	3449	3450	3452#	
MDM7B	011140	3451	3454#		
MDM7C	011150	3455	3457#		
MDM7D	011162	3458	3460#		
MDM7E	011200	3462	3464#		
MDONE	056362	13041	13058#		
MERR3	056270	13017	13027#		
MFPI0	017670	5571	5574#		
MFPI0A	017710	5576	5578#		
MFPS1	016760	5345	5348#		
MFPS2A	017032	5364	5365	5366	5368#
MFPS2B	017034	5367	5370#		
MFPS2C	017046	5371	5373#		
MFPS3A	017106	5385	5386	5387	5389#
MFPS3B	017110	5388	5391#		
MFPS3C	017122	5392	5394#		
MFPS4A	017162	5406	5407	5408	5410#

```
MFPS4B  017164     5409    5412#
MFPS4C  017176     5413    5415#
MFPS5A  017236     5427    5428     5429     5431#
MFPS5B  017240     5430    5433#
MFPS5C  017252     5434    5436#
MFPS6A  017314     5448    5449     5450     5452#
MFPS6B  017316     5451    5454#
MFPS6C  017330     5455    5457#
MFPS7A  017372     5469    5470     5471     5473#
MFPS7B  017374     5472    5475#
MFPS7C  017406     5476    5478#
MFPT   = 000007    5904#   5906
MMBBF0  106054    19177   19188    19201    19203#
MMBBF1  106064    19185*  19207#
MMBDON  106076    19195   19214#
MMBTP1  106034    19176   19198#
MMB10   106074    19183   19191    19194    19212#
MMB2    106000    19185#
MMCDON  114760    20921   20996#
MMC1    113460    20708#
MMC10   114072    20801#
MMC11   114140    20814#
MMC12   114206    20827#
MMC13   114254    20840#
MMC14   114322    20853#
MMC15   114370    20867#
MMC16   114436    20881#
MMC17   114504    20895#
MMC2    113526    20721#
MMC20   114552    20909#
MMC3    113574    20734#
MMC4    113642    20747#
MMC5    113710    20761#
MMC6    113756    20774#
MMC7    114024    20788#
MMMDON  077334    17300   17390#
MMM1    076700    17246#
MMM2    076734    17257#
MMM3    076770    17268#
MMM4    077024    17279#
MMM5    077060    17290#
MMUHLT  050520      678*  11522#
MMUTST  037374     9216    9302#
MMVEC  = 000250    9222#   9303*    9304*   10087*   10130*   10302*   10348*   10361*   10400*   10442*   10455*   10490*   10536*
                  10549*  10584*   10634*   10657*   10690*   10696*   10741*   10932*   11035*   11046*   11176*   11199*   11297*
MODDD0  100020    17516   17520    17548#
MODDD1  100030    17518   17528    17551#
MODDOV  077624    17406   17421    17435    17450    17498#
MODDSU  076500    16980   16993    17006    17019    17032    17045    17058    17071    17084    17097    17110    17123    17136
                  17182#
MODDT0  076654    17199   17203    17224#
MODDT1  076664    17201   17211    17227#
MODFD0  077314    17354   17358    17384#
MODFD1  077324    17356   17365    17387#
MODFOV  077120    17246   17257    17268    17279    17290    17336#
MODFSU  075254    16755   16764    16773    16782    16791    16800    16809    16818    16827    16836    16845    16854    16863
```

```
                          16872   16915#
MODFT0  075430            16932   16936   16955#
MODFT1  075440            16934   16943   16958#
MODP1   075450            16920   16961#  17187   17341   17503
MOR0    020376             5733    5745#
MOR1    020420             5755    5758#
MOR2    020454             5773    5776#
MOR3    020522             5793    5796#
MOR4    020534             5797    5800#
MOR5    020546             5802    5805#
MOR6    020612             5820    5823#
MOR7    020624             5825    5829#
MOR8    020636             5830    5833#
MOV1    013466             4253    4254    4256#
MOV2    013470             4255    4259#
MOV3    013506             4262    4263    4265#
MPAT10  056332            13013   13043#
MPAT11  056334            13044#
MPAT12  056336            13045#
MPAT13  056340            13046#
MPAT20  056342            13048#
MPAT21  056344            13049#
MPAT22  056346            13050#
MPAT23  056350            13051#
MRK1    016274             5181    5186#
MRK2    016316             5186    5187    5188    5190    5193#
MRK3    016320             5192    5195#
MRK4    016342             5198    5200#
MRK5    016346             5199    5202#
MRK6    016362             5203    5206#
MTFLAG  000172              637#    668*  23586
MTPI0   017756             5591    5594#
MTPS1   016406             5226    5228#
MTPS1A  016426             5232    5233    5234    5236#
MTPS2   016456             5248    5250#
MTPS3   016522             5264    5266#
MTPS4   016564             5279    5281#
MTPS5   016620             5294    5296#
MTPS6   016664             5309    5311#
MTPS7   016730             5324    5326#
MULDSU  072552            16067   16078   16089   16100   16137#
MULDT   072664            16155   16159   16171#
MULFSU  072224            15889   15897   15905   15913   15921   15929   15937   15945   15953   15961   15969   15977   15985
                          16020#
MULFT   072332            16036   16049#
M1      056230            13010#
M15     056250            13019#
M2      056254            13022#
M3      056256            13023#
M4      056260            13024#
M5      056314            13034   13036#
M6      056320            13037#  13040
M7      056326            13038   13040#
M8      056272            13026   13029#
N     = 000307             6899#   7608    7628#   7648#   7670#   7690#   7710#   7732#   7752#   7775#   7780    7803#   7826#
                           7849#   7872#   7895#   7918#   7941#   7964#   7987#   8010#   8024    8045#   8066#   8087#   8110#
```

```
                        8131#   8152#   8175#   8196#   8217#   8238#   8261#   8282#   8303#   8324#   8347#   8368#   8389#
                        8412#   8416    8437#   8458#   8479#   8500#   8521#   8542#   8563#   8584#   8605#   8626#   8632
                        8653#   8674#   8697#   8718#   8739#   8762#   8783#   8804#   8825#   8848#   8869#   8890#   8913#
                        8934#   8955#   8973#   8989#   8993    9014#   9035#   9056#   9077#   9098#   9119#   9140#   9161#
                        9182#   9203#
NATBF1   110654         19927   19933   19950   19967#
NATINS   110560         19937   19940#
NATSUB   110504         19716   19735   19754   19773   19792   19811   19830   19849   19868   19924#
NBR      001530           724*    727*    736#
NDATI0   056526         13093   13109   13119#
NDATI1   056530         13120#
NDATI2   056532         13121#
NDATI3   056534         13122#
NDATO0   056464         13086   13092   13102#
NDATO1   056466         13103#
NDATO2   056470         13104#
NDATO3   056472         13105#
NDONE    056536         13100   13124#
NEGAT    026550          6989    7017    7045    7073    7103    7132    7136#
NEG00    004570          1714    1715    1716    1718#
NEG01    004572          1717    1721#
NEG02    004600          1722    1725#
NEG03    004614          1727    1728    1729    1731#
NEG04    004616          1730    1733#
NEG1     014422          4591    4592    4593    4595#
NEG10    004644          1744    1745    1746    1748#
NEG11    004646          1747    1751#
NEG12    004650          1752    1754#
NEG13    004672          1756    1757    1758    1760#
NEG14    004674          1759    1762#
NEG2     014424          4594    4597#
NEG20    004724          1773    1774    1775    1777#
NEG21    004726          1776    1779#
NEG22    004746          1785    1787#
NEG3     014446          4601    4602    4603    4605#
NEG30    005156          1911    1912    1913    1915#
NEG31    005160          1914    1917#
NEG32    005166          1918    1920#
NEG33    005204          1923    1925#
NEG34    005212          1926    1928#
NEG4     014450          4604    4607#
NEG40    005446          2062    2063    2064    2066#
NEG41    005450          2065    2068#
NEG42    005456          2069    2071#
NEG5     014470          4611    4612    4613    4615#
NEG50    005514          2086    2087    2088    2090#
NEG51    005516          2089    2092#
NEG52    005524          2093    2095#
NEG60    005564          2110    2111    2112    2114#
NEG61    005566          2113    2116#
NEG70    005620          2128    2129    2130    2132#
NEG71    005622          2131    2134#
NERRO    056456         13079   13097#
NNBBF0   106210         19234   19236   19255#
NNBDON   106222         19244   19262#
NNBTP1   106170         19227   19247#
```

NNBTP2	106200	19237	19251#			
NNB10	106220	19240	19243	19260#		
NNB2	106116	19229#				
NNCDON	115072	21027	21036#			
NNCTB0	115054	21011	21030#			
NNCTB1	115060	21018	21021	21023	21025	21031#
NNC10	115070	21016	21022	21024	21026	21033#
NNC2	115022	21020#				
NNNDON	100040	17463	17554#			
NNN1	077340	17406#				
NNN2	077414	17421#				
NNN3	077470	17435#				
NNN4	077544	17450#				
NODL	023566	6530	6533#			
NODL2	024236	6622#				
NOP	= 000240	511	538#			
NOR	023366	6479#	6482			
NOSUB	023462	6501	6504#			
NPAT10	056516	13074	13114#			
NPAT11	056520	13115#				
NPAT12	056522	13116#				
NPAT13	056524	13117#				
NPAT20	056504	13077	13089	13109#		
NPAT21	056506	13110#				
NPAT22	056510	13111#				
NPAT23	056512	13112#				
N1	056366	13071#				
N12	056436	13090	13092#			
N13	056452	13095#	13099			
N14	056460	13096	13099#			
N2	056412	13082#				
N3	056414	13083#				
N4	056416	13084#				
ODATI0	056702	13161	13175	13186#		
ODATI1	056704	13187#				
ODATI2	056706	13188#				
ODATI3	056710	13189#				
ODATO0	056640	13153	13160	13169#		
ODATO1	056642	13170#				
ODATO2	056644	13171#				
ODATO3	056646	13172#				
ODONE	056712	13167	13191#			
OERRO	056610	13145	13158#			
OOBDON	106352	19295	19309#			
OOBTP1	106330	19273	19278	19285	19291	19298#
OOBTP2	106340	19274	19286	19302#		
OOB10	106350	19289	19292	19294	19307#	
OOB2	106264	19283#				
OOCDON	115204	21066	21075#			
OOCTB0	115166	21050	21069#			
OOCTB1	115172	21057	21060	21062	21064	21070#
OOC10	115202	21055	21061	21063	21065	21072#
OOC2	115134	21059#				
OOODON	101032	17763	17766#			
OOOT	100766	17737	17748#			
OOO2	100764	17744#	17749			

0003	101000	17753#				
0004	101024	17760	17762#			
OPAT10	056672	13140	13181#			
OPAT11	056674	13182#				
OPAT12	056676	13183#				
OPAT13	056700	13184#				
OPAT20	056656	13156	13175#			
OPAT21	056660	13143	13176#			
OPAT22	056662	13177#				
OPAT23	056664	13178#				
OPAT24	056666	13179#				
OVDNTT	073550	16430	16432	16447#		
OVDTT	074514	16717	16719	16735#		
OVFNTT	073166	16285	16287	16302#		
OVFTT	074102	16566	16568	16585#		
OVUNDN	073426	16321	16334	16347	16360	16411#
OVUNDT	074342	16604	16617	16630	16643	16692#
OVUNFN	073044	16188	16198	16207	16216	16266#
OVUNFT	073730	16466	16476	16486	16496	16542#
01	056542	13137#				
012	056612	13157	13160#			
013	056626	13163#	13166			
014	056634	13164	13166#			
02	056566	13149#				
03	056570	13150#				
04	056572	13151#				
PCN01	020652	5850#	5851			
PCN1	024512	6689#	6691			
PCN2	020670	5860#	5861			
PCN3	020720	5871#	5872			
PCN4	020744	5881#	5882			
PCN5	020766	5891#	5892			
PDATI0	057014	13210	13218	13224	13233#	
PDATI1	057016	13234#				
PDATI2	057020	13235#				
PDATI3	057022	13236#				
PDATO0	057024	13215	13219	13238#		
PDATO1	057026	13239#				
PDATO2	057030	13240#				
PDATO3	057032	13241#				
PDONE	057034	13227	13243#			
PDRTB1	044144	10440	10504#			
PDRTB2	044176	10454	10521#			
PDRTB3	044430	10534	10598#			
PDRTB4	044462	10548	10615#			
PPAT10	057004	13207	13228#			
PPAT11	057006	13229#				
PPAT12	057010	13230#				
PPAT13	057012	13231#				
PPBDON	106502	19343	19357#			
PPBTP1	106460	19320	19325	19333	19339	19346#
PPBTP2	106470	19321	19334	19350#		
PPB10	106500	19337	19340	19342	19355#	
PPB2	106414	19330#				
PPCDON	115316	21105	21113#			
PPCTB0	115300	21089	21108#			

```
PPCTB1   115304           21096   21099   21101   21103   21109#
PPC10    115314           21094   21100   21102   21104   21111#
PPC2     115246           21098#
PPPBF0   101152           17783   17815#
PPPBF1   101166           17793   17799   17803   17818#
PPPDON   101214           17813   17826#
PPPTP1   101202           17790   17804   17821#
PPP10    101212           17806   17808   17810   17812   17824#
PPP2     101052           17785#  17786
PPP3     101102           17797#
PPP4     101114           17800   17803#
PS     = 177776             514#   1393*   1394    1402*   1403    1411*   1412    1420*   1421    5208#   5225    5245*   5247
                           5261*   5263    5276*   5278    5293    5306*   5308    5321*   5323    5342*   5350*   5362*   5383*
                           5404*   5425*   5446*   5467*   5499*   5501    5515*   5521*   5523*   5537*   5541*   5543*   5548*
                           5550*   5554*   5567*   5570    5572*   5574*   5584*   5586*   5590    5592*   5594*   5654*   5655*
                           5656    5659*   5661*   5706    5722    5933*   5936*   5945*   5951*   9277   21857*  23372*  23413*
                          23444*
PSW    = 177776            9277#   9339*   9340*   9341    9351*   9357*   9359*   9360    9369*   9371*   9372    9385*   9387*
                           9389*   9522*   9532*   9917    9939   10070*  10094   10111   10121*  10123*  10193*  10238*  10306
                          10339*  10345*  10347*  10365   10397*  10399*  10750*  10755*  10767*  10770*  10772*  10782*  10786
                          10788*  10847*  10856*  10858*  10860*  10936*  10952*  10963*  10974*  10984*  10997*  11010*  11022*
                          11049*  11060*  11079*  11094*  11110*  11126*  11143*  11159*  11190*  11201*  11215*  11227*  11238*
                          11248*  11260*  11273*  11285*  11298*  11304*  11335*  11369   11371   11601
PSWORD   025416            6903#   6904    6969*   6970    6998*   6999    7026*   7027    7054*   7055    7083*   7084    7113*
                           7114    7212*   7213    7229*   7230    7246*   7247    7263*   7264    7280*   7281    7297*   7298
                           7314*   7315    7331*   7332    7348*   7349    7365*   7366    7406*   7407    7443*   7444    7480*
                           7481    7617*   7618    7637*   7638    7658*   7659    7679*   7680    7699*   7700    7720*   7721
                           7741*   7742    7761*   7762    7789*   7790    7812*   7813    7835*   7836    7858*   7859    7881*
                           7882    7904*   7905    7927*   7928    7950*   7951    7973*   7974    7996*   7997    8031*   8032
                           8052*   8053    8073*   8074    8095*   8096    8117*   8118    8138*   8139    8160*   8161    8182*
                           8183    8203*   8204    8224*   8225    8246*   8247    8268*   8269    8289*   8290    8310*   8311
                           8332*   8333    8354*   8355    8375*   8376    8397*   8398    8423*   8424    8444*   8445    8465*
                           8466    8486*   8487    8507*   8508    8528*   8529    8549*   8550    8570*   8571    8591*   8592
                           8612*   8613    8640*   8641    8661*   8662    8683*   8684    8705*   8706    8726*   8727    8748*
                           8749    8770*   8771    8791*   8792    8812*   8813    8834*   8835    8856*   8857    8877*   8878
                           8899*   8900    8921*   8922    8942*   8943    8964*   8965*   8966    8981*   8982*   8983    9001*
                           9002    9022*   9023    9043*   9044    9064*   9065    9085*   9086    9106*   9107    9127*   9128
                           9148*   9149    9169*   9170    9190*   9191
PUSRM  = 030000             518#
PWRDN    131574             663   23581#  23584
PWRMSG   131674           23588   23608#
PWRUP    131604           23581   23584#
P1       056716           13204#
P2       056732           13212#  13213
P3     = 056734           13213#
P4       056736           13215#
P5       056760           13220#  13223
QDATI0   057156           13272   13288   13298#
QDATI1   057160           13299#
QDATI2   057162           13300#
QDATI3   057164           13301#
QDATO0   057146           13267   13271   13293#
QDATO1   057150           13294#
QDATO2   057152           13295#
QDATO3   057154           13296#
QDONE    057166           13281   13303#
```

```
QPAT10  057126      13259   13283#
QPAT11  057130      13284#
QPAT12  057132      13285#
QPAT13  057134      13286#
QPAT20  057136      13262   13278   13288#
QPAT21  057140      13289#
QPAT22  057142      13290#
QPAT23  057144      13291#
QQBDON  106644      19390   19406#
QQBTP1  106612      19367   19372   19380   19386   19393#
QQBTP2  106632      19368   19381   19399#
QQB10   106642      19384   19387   19389   19404#
QQB2    106546      19377#
QQCDON  115440      21144   21153#
QQCTB0  115416      21127   21147#
QQCTB1  115422      21135   21140   21142   21148#
QQCTB2  115432      21134   21138   21142   21150#
QQC10   115436      21132   21139   21141   21143   21151#
QQC2    115364      21137#
QQQBF0  101334      17844   17874#  17888
QQQBF1  101350      17854   17859   17863   17876#  17897   17900   17903
QQQDON  101502      17909   17913#
QQQTP1  101364      17851   17864   17878#  17895   17904
QQQ10   101374      17866   17868   17870   17872   17883#
QQQ2    101234      17846#  17847
QQQ20   101376      17873   17888#
QQQ22   101410      17891#  17892
QQQ23   101442      17898   17899#
QQQ24   101454      17901   17903#
QQQ25   101500      17907   17911#
QQQ3    101264      17857#
QQQ4    101276      17861   17863#
Q1      057040      13256#
Q2      057054      13264#  13265
Q3    = 057056      13265#
Q4      057060      13267#
Q5      057102      13273#  13276
RBUF    120534      21950#  22059   22249   22283   22316   22345   22376   22392   22438   22441   22447   22461   23459
                    23464
RBUF2   124464      22747#  22856   23065   23092   23102   23130   23156   23185   23200   23248   23251   23257   23272
                    23281   23297   23300   23303   23318   23476   23481
RCSR    120532      21949#  22046   22127   22131*  22132   22135*  22136   22140*  22142   22259   22268   22281   22285
                    22296*  22302   22310*  22313*  22328   22330*  22335*  22359   22362*  22369*  22386*  22389   22395*
                    22410*  22413   22416*  22419*  22459   23436*
RCSR2   124462      22746#  22843   22934   22938*  22939   22942*  22943   22951*  22953   23074   23086   23100   23103
                    23114*  23119   23125*  23128*  23142   23144*  23149*  23170   23172*  23179*  23196*  23198   23203*
                    23223*  23225   23228*  23231*  23270   23275   23294   23313   23439*
RECCT1  130376      23421*  23458*  23496   23527#
RECCT2  130400      23422*  23475*  23507   23528#
REC1    130054      23426   23458#
REC2    130136      23430   23475#
REGR1   025664       6988    6991#
REGR2   026012       7016    7019#
REGR23  030106       7431    7434#
REGR3   026140       7044    7047#
REGR4   026264       7072    7075#
```

REGR5	026414	7102	7105#		
REG01	027726	7396#	7508		
REG1	003422	1195#	1198		
REG1A	003450	1210#	1213		
REG1E	003434	1196	1200#		
REG2	003502	1225#	1228		
REG2B	003530	1240#	1243		
REG2E	003514	1226	1230#		
REG3	003562	1256#	1259		
REG3A	003610	1271#	1274		
REG3E	003574	1257	1261#		
REG4	003642	1287#	1290		
REG4A	003670	1302#	1305		
REG4E	003654	1288	1292#		
REG45	030262	7467	7470#		
REG5	003722	1319#	1322		
REG5A	003750	1334#	1337		
REG5E	003734	1320	1324#		
REG6	004002	1350#	1353		
REG6A	004030	1365#	1368		
REG6E	004014	1351	1355#		
RESET2	024306	6633	6641#		
RESET3	024304	6639#			
REST	017456	5504#			
RESTRT	001206	663#	23572	23590	
RET	025214	6826	6834#		
RETA	050450	11501	11503#		
RETAT	022744	6335	6341#		
RETBT	022774	6347	6353#		
RETCT	023030	6361	6366#		
RETR1	024520	6688	6691#		
RETR2	024556	6703	6707#		
RETR3	024602	6720	6724#		
RET1	025224	6835	6837#		
RET2	025236	6838	6840#		
RET3	025246	6841	6844#		
RET4	025244	6842#			
RITSH	030446	7432	7468	7506	7509#
ROL1	015076	4785	4786	4787	4789#
ROL2	015100	4788	4791#		
ROL3	015116	4794	4795	4796	4798#
ROL4	015120	4797	4800#		
ROL5	015134	4803	4804	4806#	
ROL6	015136	4805	4808#		
ROL7	015160	4811	4812	4813	4816#
RORB1	015536	4956	4958#		
RORB10	015620	4982	4983	4985#	
RORB11	015622	4984	4987#		
RORB12	015634	4988	4990#		
RORB13	015646	4991	4992	4993	4995#
RORB14	015650	4994	4997#		
RORB2	015550	4959	4960	4961	4963#
RORB3	015552	4962	4965#		
RORB4	015562	4966	4967	4969#	
RORB5	015564	4968	4971#		
RORB7	015574	4972	4974#		

```
ROR1     015204     4826    4827    4828    4830#
ROR2     015206     4829    4832#
ROR3     015224     4835    4836    4837    4839#
ROR4     015226     4838    4841#
ROR5     015244     4844    4845    4846    4848#
ROR6     015246     4847    4850#
ROR7     015262     4853    4854    4856#
ROTX     012160     3768    3770*   3774    3780#   3793*   3795*   3797    3813*   3814    3818
ROTXAD   012250     3814*   3816*   3822#
ROT0A    011464     3586    3587    3590#
ROT0B    011466     3589    3592#
ROT0C    011510     3595    3596    3599#
ROT1A    011540     3619    3620    3623#
ROT1B    011542     3622    3625#
ROT1C    011566     3628    3629    3632#
ROT1D    011570     3631    3634#
ROT1E    011620     3639    3640    3643#
ROT2A    011654     3661    3663    3667#
ROT2B    011656     3666    3669#
ROT2C    011706     3673    3675    3678#
ROT2D    011710     3677    3680#
ROT2E    011744     3685    3687    3691#
ROT3A    011774     3708    3711#
ROT3B    011776     3710    3713#
ROT3C    012024     3716    3719#
ROT3D    012026     3718    3721#
ROT3E    012054     3724    3727#
ROT4     012112     3746    3748    3751#
ROT5     012156     3773    3775    3778#
ROT6     012210     3796    3799#
ROT7     012246     3817    3820#
RPSW     120544     21954#  22353   22355*  22371*  23427*
RPSW2    124474     22751#  23165   23167*  23181*  23431*
RRBDON   107004     19442   19457#
RRBTP1   106760     19418   19424   19425*  19432   19445#  19453
RRBTP2   106770     19419   19433   19449#
RRBTP3   107000     19423   19438   19453#
RRB10    107002     19436   19439   19441   19455#
RRB2     106714     19429#
RRCDON   115564     21185   21194#
RRCTB0   115542     21168   21188#
RRCTB1   115546     21176   21181   21183   21189#
RRCTB2   115556     21175   21179   21183   21191#
RRC10    115562     21173   21180   21182   21184   21192#
RRC2     115510     21178#
RRRDON   101654     17957   17975#
RRREXP   101644     17948   17971#
RRRTP1   101624     17926   17961#
RRRTP2   101634     17932   17966#
RRR10    101622     17934   17952   17955   17958#
RRR2     101554     17943#
RRR3     101556     17925   17944#  17949
RRR4     101602     17951#  17953
RTCPSW   120556     21961#  22570   22572*  22602*  22618   22620*  22639*  23433*
RTCVT    120554     21960#  22508   22509*  22533*  22569   22571*  22582*  22590*  22601*  22617   22619*  22631*  22638*
                    22652   22653*  22665*  22673   22674*  22687*  23418   23432*  23522*
```

```
RTI1      023200     6414    6418#
RTI2      023204     6415    6420#
RTRAP =   000010      541#    763*    764*    778*    779*   6742*
RTRAP1=   000034      532#    792*    793*    807*    808*
RTRAP2=   000020      531#    838*    839*    853*    854*
RTRAP3=   000030      530#    867*    868*    882*    883*
RTRAP4=   000014      529#    904*    905*    919*    920*   6335*   6347*   6361*
RTRAP5=   000004      528#    939*    940*    954*    955*    968*    969*    983*    984*   6688*   6703*   6720*   6740*
RTS1      013440     4224    4229#
RTT1      023070     6379    6382#
RTT2      023076     6380    6386#
RTT3      023130     6394    6398#
RTT4      023140     6395    6402#
RTT5      023102     6392#   6400    6405
RTT6      023152     6403    6407#
RVECT     120542    21953#  22124   22125*  22145*  22298   22299*  22309*  22315*  22324   22325*  22336*  22344*  22352
                    22354*  22366*  22370*  22375*  22384   22385*  22399*  22408   22409*  22423*  23417   23426*  23521*
RVECT2    124472    22750#  22931   22932*  22956*  23115   23116*  23124*  23131*  23139   23140*  23150*  23157*  23164
                    23166*  23176*  23180*  23186*  23194   23195*  23207*  23221   23222*  23235*  23415   23430*  23519*
R0TRAP    023412     6483    6488#
R1ERR     003466     1211    1216#
R2ERR     003546     1241    1246#
R3ERR     003626     1272    1277#
R4ERR     003706     1303    1308#
R5ERR     003766     1335    1340#
R6ERR     004046     1366    1371#
SADR      055472    12767*  12778   12783*  12786#
SBC1      014770     4733    4734    4735    4737#
SBC2      014772     4736    4739#
SBC3      015010     4742    4743    4744    4746#
SBC4      015012     4745    4748#
SBC5      015030     4751    4752    4753    4755#
SBC6      015032     4754    4757#
SBC7      015052     4761    4762    4764#
SB0       012264     3838    3840#
SB2       012344     3878    3880#
SB4       012426     3918    3920#
SB5       012470     3946#
SB5A      012462     3942    3944#
SB5X      012472     3939*   3941    3948#   3949
SB5XAD    012474     3938    3944    3949#
SB6       012524     3970#
SB6X      012526     3965*   3966    3972#
SB7       012556     3994#
SB7X      012560     3989*   3996#   3997
SB7XAD    012562     3990    3997#
SCOPE =   000240      511#
SC3       020322     5702*   5721#   5730*   5731
SC4       020336     5703*   5725#   5729*
SDAT00    055506    12725   12728   12751   12754   12793#
SDAT01    055510    12794#
SDAT02    055512    12795#
SDAT03    055514    12796#
SDONE     055516    12783   12798#
SECND     124410    22728#
SERR0     055402    12712   12764#
```

SERR1	055310	12717	12735#	12772	12777	
SERR10	055420	12768#	12784			
SERR4	055452	12742	12780#			
SETBR	001410	714#	751			
SETCC	001430	718#	746			
SETCD	020320	5717	5720#	5732		
SETREG	050152	9715	9754	11406#		
SETUP	001372	711#				
SET2BR	001500	723	727#			
SHL	003234	1097#	1100			
SHLE	003250	1098	1102#			
SHR	003314	1123#	1126			
SHRE	003330	1124	1128#			
SKTST2	024324	6644#				
SLU1ST	120560	6534	6554	11548	21902	21963#
SLU2ST	124502	22487	22698	22724	22755#	
SL1HLT	123216	681*	22482#			
SL2HLT	127302	682*	23343#			
SNMB0A	005736	2206	2207	2209#		
SNMB1A	006006	2253	2254	2256#		
SNMB1B	006010	2255	2258#			
SNMB1C	006032	2263	2264	2265	2267#	
SNMB2A	006112	2315	2316	2318#		
SNMB2B	006114	2317	2320#			
SNMB2C	006122	2321	2323#			
SNMB2D	006142	2327	2328	2329	2331#	
SNMB2E	006144	2330	2333#			
SNMB3A	006244	2389	2390	2392#		
SNMB3B	006246	2391	2394#			
SNMB3C	006264	2397	2398	2399	2401#	
SNMB3D	006266	2400	2403#			
SNM0A	005714	2182	2183	2184	2186#	
SNM1A	005762	2229	2230	2231	2233#	
SNM2A	006056	2286	2287	2288	2290#	
SNM2B	006060	2289	2292#			
SNM3A	006202	2356	2357	2358	2360#	
SNM3B	006204	2359	2362#			
SNM4A	006320	2425	2426	2428#		
SNM4B	006322	2427	2430#			
SNM5A	006356	2453	2454	2456#		
SNM5B	006360	2455	2458#			
SNM6A	006416	2481	2482	2484#		
SNM6B	006420	2483	2486#			
SNM7A	006454	2508	2509	2511#		
SNM7B	006456	2510	2513#			
SOB1	016210	5152#	5159			
SOB2	016216	5152	5153	5155#		
SOB3	016220	5154	5157#			
SOB4	016240	5160	5161	5162	5165#	
SOPA	005650	2155	2157#			
SOPB	005670	2159	2162#			
SOPB0A	004244	1513	1515#			
SOPB0B	004254	1516	1519#			
SOPB1A	004316	1561	1563#			
SOPB1B	004334	1564	1567	1570#		
SOPB1C	004354	1592	1594#			

```
SOPB1D  004374          1599    1602#
SOPB2A  004460          1658    1660#
SOPB2B  004500          1664    1668#
SOPB2C  004524          1687    1689#
SOPB2D  004550          1695    1699#
SOPB3A  005034          1848    1850#
SOPB3B  005060          1853    1855    1860#
SOPB3C  005102          1886    1888#
SOPB3D  005124          1894    1897#
SOPX    005642          2154#   2157*   2158*   2164
SOPXAD  005672          2160*   2164#
SOPZA   004416          1625    1627#
SOP0A   004160          1460    1462#
SOP0B   004172          1465    1467#
SOP0C   004210          1487    1489#
SOP0D   004234          1491    1494    1499#
SOP1A   004266          1536    1538#
SOP1B   004300          1539    1543#
SOP2B   004436          1630    1635#
SOP3A   004774          1814    1816#
SOP3B   005010          1819    1822#
SOP4A   005244          1950    1952#
SOP4B   005264          1955    1960#
SOP5A   005316          1991    1993#
SOP5B   005332          1996    1999#
SOP6A   005352          2018    2020#
SOP6B   005366          2021    2024#
SOP7A   005410          2045    2047#
SOP7B   005424          2048    2051#
SPAT10  055476          12709   12729   12740   12755   12788#
SPAT11  055500          12789#
SPAT12  055502          12790#
SPAT13  055504          12791#
SR0   = 177572          9226#   9306*   9415    9476    9523*   9525    9531*   9851*   9853*   9875*   9877*   9899*   9901*
                        9924*   9926*   9945*   9947*   9975*   9977*   9999*   10001*  10023*  10025*  10047*  10049*  10089*
                        10132*  10248   10249*  10320   10335*  10378   10393*  10465   10481*  10484*  10559   10575*  10578*
                        10648*  10694   10699   10715*  10716   10730*  10812   10828*  10853   10855*  11356*
SR1   = 177574          9227#   9539
SR2   = 177576          9228#   9485    9505    10321   10379   10466   10560   10641   10674   10682   10695   10700   10720
                        10731   10813   10854
SR3   = 172516          9229#   9545*   9546    9553*   9554    9850*   9874*   9898*   9923*   9944*   9974*   9998*   10022*
                        10046*  10088*
SSBDON  107146          19492   19507#
SSBTP1  107122          19468   19474   19475*  19482   19495#  19503
SSBTP2  107132          19469   19483   19499#
SSBTP3  107142          19473   19488   19503#
SSB10   107144          19486   19489   19491   19505#
SSB2    107056          19479#
SSCDON  115710          21227   21235#
SSCTB0  115672          21208   21230#
SSCTB1  115676          21216   21221   21223   21225   21231#
SSC10   115706          21213   21220   21222   21224   21226   21233#
SSC2    115630          21218#
SSSA1   102016          17997   18010   18022#
SSSBF0  102006          17988   18001   18003   18018#
SSSDON  102040          18015   18035#
```

SSSTP1	102026	17994	18004	18028#										
SSS10	102036	17996	18002	18007	18013	18033#								
SSS2	101726	17999#												
START	001200	654	662#											
STATUS=	177776	539#	765*	769	780*	783	794*	798	809*	812	840*	844	855*	858
		869*	873	884*	887	906*	910	921*	924	941*	945	956*	959	970*
		974	985*	988	6488*	6489	6516*	6517	6547*	6553*	6567*	6574*	6588*	6641*
		6667*	6670											
STBOT	001000	545	546	651#	655	664	671	762	771	773	775	785	791	800
		802	804	814	837	846	848	850	860	866	875	877	879	889
		903	912	914	916	926	938	947	949	951	961	967	976	978
		980	990	1392	4116	4131	4146	4152	4222	5178	5191	5202	5553	5555
		5566	5575	5588	5595	5905	6149	6313	6334	6346	6353	6360	6366	6376
		6392	6412	6430	6449	6455	6478	6492	6505	6520	6529	6566	6587	6632
		6656	6673	6687	6691	6758	6774	6779	6827	6834	6837	6840	6940	9302
		11545	21818	21926	23585									
STCDFS	103606	18428	18447	18466	18485	18504	18554#							
STCDT	103736	18560	18566	18573	18589#									
STCFDS	103154	18248	18267	18286	18305	18324	18378#							
STCFT	103304	18384	18390	18397	18413#									
STCIBF	117272	21559	21565	21572	21590#									
STCSUB	117144	21347	21357	21367	21377	21388	21398	21408	21418	21429	21439	21449	21459	21469
		21479	21489	21500	21511	21554#	21608							
STP	022702	6325	6328#											
STP3	023336	6457#												
STP3D	023340	6456	6459#											
STP4	024462	6675#												
STP4E	024464	6653	6674	6677#										
STXBF	117666	21722*	21723*	21730	21733	21740	21746#							
STXSUB	117554	21631	21640	21649	21659	21669	21679	21720#						
SUB0	006540	2563	2564	2565	2567#									
SUB0A	006542	2566	2569#											
SUB1	014656	4692	4693	4695#										
SUB2	014660	4694	4697#											
SUB3	014702	4701	4702	4704#										
SUB4	014704	4703	4706#											
SUB5	014722	4710	4711	4713#										
SUB6	014724	4712	4715#											
SUB7	014744	4718	4719	4720	4722#									
SWB1	014154	4472	4473	4475#										
SWB2	014156	4474	4477#											
SWB3	014174	4480	4481	4482	4484#									
SWR	021324	6030#	6034*	6796*	6811*	6818	9307	11546	21963	22485	22755	23357		
SWREG	000176	639#												
SXT0	016056	5079	5080	5081	5082	5085#								
SXT1	016060	5084	5087#											
SXT2	016110	5093	5094	5095	5096	5099#								
SX10	055344	12749#												
SX11	055370	12757#	12760											
SX12	055376	12758	12760#											
SX2	055246	12720#												
SX3	055250	12721#	12764											
SX4	055254	12723#												
SX5	055300	12731#	12734											
SX6	055306	12732	12734#											
SX7	055312	12739#	12767											

```
SX8      055336     12746#
SX9      055340     12747#  12780
S1       025434      6915#   6916    7204    7245    7777    7788    7834
S10      025456      6924#   8991    9021
S11      025460      6925#
S2       025436      6916#   7205    7228    7778    7811
S3       025440      6917#
S4       025442      6918#
S5       025444      6919#   6920    8413    8422    8464
S6       025446      6920#   8414    8443
S7       025450      6921#
S8       025452      6922#
S9       025454      6923#   6924    8990    9000    9042
TABLE    025260      6813    6848#
TBIT   = 000020      9278#  11369   11374   11387
TBITPS   037360      9293#   9305*  11372*  11387   11389   11390*
TBLEND   127324     23354#  23388
TBL1     011256      3484    3485*   3486*   3487*   3488*   3489    3498#   3514*   3515*   3516*   3517*   3518    3522
                     3523    3524    3525    3526    3541    3542*   3543*   3544*   3545*   3546    3564*   3565*   3566*
                     3567*   3568
TBL2     011324      3513    3526#   3563
TBUF     120540     21952#  21990   22006*  22011*  22028*  22031*  22235*  22238*  22258*  22280*  22301*  22327*  22358*
                    22388*  22412*  22432*  22458*  23452*
TBUF2    124470     22749#  22782   22798*  22803*  22825*  22828*  23051*  23054*  23073*  23099*  23118*  23141*  23169*
                    23197*  23224*  23243*  23269*  23293*  23311*  23469*
TCCBF0   111660     20253   20261   20271#
TCCBF1   111670     20257   20273#
TCCDON   111700     20266   20276#
TCC2     111620     20254#
TCC3     111650     20262   20265#
TCSR     120536     21951#  21976   22007   22012   22016   22029   22034   22074   22077*  22078   22081*  22082   22085*
                    22087   22099   22103*  22104   22107*  22108   22112*  22114   22153*  22156   22163*  22168*  22177*
                    22181   22183*  22188*  22202*  22208   22210*  22219*  22230*  22234*  22236   22241*  22256*  22263*
                    22279*  22284*  22295*  22297*  22304*  22307*  22326*  22333*  22339*  22342*  22357*  22361*  22387*
                    22391*  22411*  22430*  22433   22437*  22456*  22464*  22468*  23434*  23435*  23455*  23486*
TCSR2    124466     22748#  22768   22799   22804   22808   22826   22831   22871   22874*  22875   22878*  22879   22888*
                    22890   22902   22906*  22907   22910*  22911   22919*  22921   22964*  22967   22974*  22979*  22988*
                    22992   22994*  22999*  23013*  23019   23021*  23030*  23046*  23050*  23052   23057*  23113*  23244
                    23273*  23292*  23296*  23438*  23472*  23487*
TDATI0   052502     12010   12015   12030   12040   12045   12071   12095#
TDATI1   052504     12096#
TDATI2   052506     12068*  12097#
TDATI3   052510     12069*  12098#
TDATO0   052472     12026   12029   12065   12070   12090#
TDATO1   052474     12091#
TDATO2   052476     12092#
TDATO3   052500     12093#
TDEC1    022162      6190    6193#
TDEC2    022176      6200    6202#
TDEC3    022242      6215    6217#
TDEC4    022274      6222    6225#
TDEC6    022310      6228#
TDEC7    022312      6226    6230#
TDEC77   023704      6549    6558#
TDEC8    023702      6551    6556#
TDONE    052512     12077   12100#
```

```
TEMP1    025420       6550*   6561    6568*   6581    6591*   6603    6657*   6677    6905#   6906    6941*   6961    6991
                      7019    7047    7075    7106    7136*   7142*   7157*   7164*   7182*   7195*   7389*   7397    7434
                      7471    7510*   7518*   7532*   7540*   7547*   7552*   7558*   7589*   7596*
TEMP2    025422       6907#   6908    6942*   6964    6968    6985*   6994    6997    7013*   7022    7025    7041*   7050
                      7053    7069*   7079    7082    7099*   7109    7112    7129*   7137*   7143*   7149*   7153*   7158*
                      7165*   7170*   7176*   7183*   7189*   7390*   7398    7435    7472    7511*   7559*   7566*   7577*
                      7582*   7590*   7597*
TEMP3    025424       6909#   6910    6943*   6974    6986*   7003    7014*   7031    7042*   7059    7070*   7088    7100*
                      7118    7130*   7138*   7144*   7159*   7166*   7171*   7177*   7184*   7190*   7391*   7402    7405
                      7426*   7439    7442    7462*   7476    7479    7500*   7512*   7524*   7528*   7533*   7553*   7560*
                      7571*   7578*   7583*   7598*
TEMP4    025426       6911#   6912    6944*   6970    6999    7027    7055    7084    7114    7145*   7160*   7172*   7178*
                      7185*   7191*   7196*   7392*   7411    7429*   7448    7465*   7485    7503*   7513*   7519*   7534*
                      7541*   7567*   7572*   7584*   7599*
TEMP5    025430       6913#   7393*   7414    7428*   7451    7464*   7488    7502*   7514*   7535*   7542*   7548*   7561*
                      7585*   7591*
TEMP6    025432       6914#   7394*   7407    7444    7481    7520*   7536*   7543*   7554*   7562*   7573*   7592*   7600*
TENSAV   025044       6788*   6801#   6812
TESTN1   021326       5953    6032#
TEST1    014112       4449    4450    4451    4453#
TEST2    014114       4452    4455#
TEST3    014132       4459    4460    4462#
THRPRT   025466       6831    6935#
TICKER   130010      23432   23448#
TICKS    130402      23423*  23448*  23483   23529#
TKB    = 177562        535#   6616
TOFF     050064      10144   10194   10245   10749   10838   11369#
TOLER    124376      22721   22723#
TON      050120      10187   10237   10272   10774   10892   11387#
TONT1    023262       6434    6443#
TO10     021260        616    1007    5999#   6241    6760    6780    6830   21924
TO114    021272        628    6009#
TO14     021262        618    1008    6001#   6283    6443    6459    6643
TO20     021264        620    1010    6003#   6251    6444
TO244    021274        631    6011#
TO250    021276        633    6013#   9303   10130   10348   10400   10490   10584   10657   10741   10830   10890   11035
                     11176   11297
TO30     021266        623    6005#
TO34     021270        625    1009    6007#   6273    6602
TO4      021256        614     669    1006    5997#   6293    6306    6328    6524    6533    6535    6559    6759   10271
                     10773   10829   10888   21923
TPAT10   052452      12009   12039   12080#
TPAT11   052454      12081#
TPAT12   052456      12082#
TPAT13   052460      12083#
TPAT20   052462      12018   12048   12085#
TPAT21   052464      12086#
TPAT22   052466      12087#
TPAT23   052470      12088#
TPB    = 177566        516#    536#   6662*   6665*  21858*  21861*  21872*  21875*  23597*
TPS    = 177564        515#
TPSW     120550      21956#  22204   22206*  22224*  23425*
TPSW2    124500      22753#  23015   23017*  23035*  23429*
TRACE    023256       6431    6439#
TRAPA  = 000077        540#    766     781    6745
TRAPB    023534       6493    6511    6524#
```

```
TRAPC = 104777          548#    810
TRAPPC  037350         9289#
TRAPPS  037352         9290#
TRAP10  025046         6789    6802#
TRCSR = 177560          534#   6617*   6619
TRC1    023326         6450    6453    6455#
TRPADR  023410         6485#   6492
TRT   = 000003          526#    907     922
TR0     024014         6569    6576#
TR2     024020         6572    6580#
TR3     024126         6590    6597#
TR4     024132         6592    6600#
TR5     024130         6594    6598#
TSTSPC  037310         9207#
TST160  030502         7433    7469    7507    7516#
TST161  030530         7517    7522#
TST162  030546         7523    7526#
TST163  030564         7527    7530#
TST164  030626         7531    7538#
TST165  030664         7539    7545#
TST166  030710         7546    7550#
TST167  030740         7551    7556#
TST170  030776         7557    7564#
TST171  031022         7565    7569#
TST172  031050         7570    7575#
TST173  031072         7576    7580#
TST174  031130         7581    7587#
TST175  031164         7588    7594#
TST176  031236         7612#
TST177  031304         7632#
TST200  031352         7652#
TST201  031424         7674#
TST202  031472         7694#
TST203  031540         7714#
TST204  031612         7736#
TST205  031660         7756#
TST206  031752         7784#
TST207  032030         7807#
TST210  032106         7830#
TST211  032164         7853#
TST212  032240         7876#
TST213  032314         7899#
TST214  032370         7922#
TST215  032446         7945#
TST216  032524         7968#
TST217  032600         7991#
TST220  032654         8028#
TST221  032724         8049#
TST222  032774         8070#
TST223  033044         8091#
TST224  033120         8114#
TST225  033170         8135#
TST226  033240         8156#
TST227  033314         8179#
TST230  033364         8200#
TST231  033434         8221#
```

```
TST232  033504     8242#
TST233  033560     8265#
TST234  033630     8286#
TST235  033700     8307#
TST236  033750     8328#
TST237  034024     8351#
TST240  034074     8372#
TST241  034144     8393#
TST242  034234     8420#
TST243  034304     8441#
TST244  034354     8462#
TST245  034424     8483#
TST246  034472     8504#
TST247  034540     8525#
TST250  034606     8546#
TST251  034656     8567#
TST252  034726     8588#
TST253  034774     8609#
TST254  035042     8636#
TST255  035116     8657#
TST256  035172     8678#
TST257  035252     8701#
TST260  035326     8722#
TST261  035402     8743#
TST262  035462     8766#
TST263  035536     8787#
TST264  035612     8808#
TST265  035666     8829#
TST266  035746     8852#
TST267  036022     8873#
TST270  036076     8894#
TST271  036156     8917#
TST272  036232     8938#
TST273  036306     8959#
TST274  036354     8977#
TST275  036432     8997#
TST276  036506     9018#
TST277  036562     9039#
TST300  036636     9060#
TST301  036710     9081#
TST302  036762     9102#
TST303  037034     9123#
TST304  037110     9144#
TST305  037164     9165#
TST306  037236     9186#
TST37   026576     6990    7018    7046    7074    7104    7134    7140#
TST40   026632     7141    7147#
TST41   026646     7148    7151#
TST42   026664     7152    7155#
TST43   026720     7156    7162#
TST44   026752     7163    7168#
TST45   027004     7169    7174#
TST46   027036     7175    7180#
TST47   027074     7181    7187#
TST50   027124     7188    7193#
TST51   027176     7210#
```

TST52	027236	7227#	
TST53	027276	7244#	
TST54	027336	7261#	
TST55	027374	7278#	
TST56	027432	7295#	
TST57	027470	7312#	
TST60	027530	7329#	
TST61	027570	7346#	
TST62	027626	7363#	
TS1	001372	676	710#
TS10	002646	966#	
TS100	005716	2185	2200#
TS101	005740	2208	2223#
TS102	005764	2232	2246#
TS103	006034	2266	2280#
TS104	006070	2294	2308#
TS105	006154	2335	2348#
TS106	006214	2364	2379#
TS107	006276	2405	2418#
TS11	003042	1016#	
TS110	006330	2431	2444#
TS111	006370	2460	2473#
TS112	006426	2487	2500#
TS113	006464	2514	2526#
TS114	006502	2532	2542#
TS115	006520	2548	2558#
TS116	006550	2570	2584#
TS117	006644	2613	2624#
TS12	003060	1020	1026#
TS120	006710	2639	2650#
TS121	006766	2672	2682#
TS122	007026	2697	2710#
TS123	007046	2717	2729#
TS124	007066	2736	2749#
TS125	007106	2756	2773#
TS126	007142	2786	2799#
TS127	007164	2807	2820#
TS13	003100	1030	1036#
TS130	007212	2830	2845#
TS131	007242	2854	2866#
TS132	007274	2878	2891#
TS133	007320	2896	2908#
TS134	007344	2913	2925#
TS135	007370	2930	2936#
TS136	007422	2948	2954#
TS137	007454	2966	2972#
TS14	003120	1040	1046#
TS140	007520	2988	2994#
TS141	007614	3021	3028#
TS142	007714	3056	3062#
TS143	007770	3079	3085#
TS144	010064	3110	3116#
TS145	010142	3133	3139#
TS146	010216	3155	3161#
TS147	010274	3177	3189#
TS15	003140	1049	1069#

TS150	010324	3201	3213#
TS151	010366	3230	3241#
TS152	010454	3263	3274#
TS153	010564	3296	3309#
TS154	010626	3323	3337#
TS155	010714	3358	3373#
TS156	011010	3395	3409#
TS157	011112	3431	3444#
TS16	003166	1074	1080#
TS160	011212	3465	3483#
TS161	011260	3490	3512#
TS162	011326	3519	3540#
TS163	011374	3547	3562#
TS164	011442	3569	3582#
TS165	011512	3598	3614#
TS166	011622	3642	3656#
TS167	011746	3690	3704#
TS17	003220	1087	1093#
TS170	012056	3726	3741#
TS171	012114	3750	3767#
TS172	012162	3777	3792#
TS173	012212	3798	3812#
TS174	012252	3819	3835#
TS175	012274	3841	3854#
TS176	012320	3859	3873#
TS177	012354	3881	3895#
TS2	001614	761#	
TS20	003252	1101	1108#
TS200	012400	3899	3913#
TS201	012434	3921	3937#
TS202	012476	3945	3964#
TS203	012530	3969	3988#
TS204	012564	3993	4034#
TS205	013032	4089	4112#
TS206	013420	4206	4221#
TS207	013450	4230	4249#
TS21	003300	1113	1119#
TS210	013510	4264	4270#
TS211	013554	4286	4292#
TS212	013620	4307	4313#
TS213	013666	4330	4348#
TS214	013752	4374	4381#
TS215	014054	4414	4431#
TS216	014074	4438	4445#
TS217	014134	4461	4467#
TS22	003332	1127	1154#
TS220	014176	4483	4500#
TS221	014314	4537	4544#
TS222	014400	4569	4586#
TS223	014472	4614	4621#
TS224	014612	4655	4662#
TS225	014634	4669	4687#
TS226	014746	4721	4728#
TS227	015054	4763	4780#
TS23	003344	1158	1164#
TS230	015162	4815	4821#

TS231	015264	4855	4861#	
TS232	015374	4897	4903#	
TS233	015514	4942	4950#	
TS234	015662	4998	5005#	
TS235	015730	5022	5030#	
TS236	016030	5060	5074#	
TS237	016112	5098	5112#	
TS24	003360	1167	1173#	
TS240	016200	5135	5148#	
TS241	016242	5164	5177#	
TS242	016364	5205	5221#	
TS243	016430	5235	5242#	
TS244	016466	5251	5257#	
TS245	016532	5267	5273#	
TS246	016572	5282	5288#	
TS247	016630	5297	5303#	
TS25	003374	1176	1182#	
TS250	016674	5312	5318#	
TS251	016740	5327	5341#	
TS252	017006	5353	5359#	
TS253	017056	5374	5380#	
TS254	017132	5395	5401#	
TS255	017206	5416	5422#	
TS256	017262	5437	5443#	
TS257	017340	5458	5464#	
TS26	003410	1185	1191#	
TS260	017416	5479	5493#	
TS261	017456	5497	5502	5514#
TS262	017516	5536#		
TS263	017630	5565#		
TS264	017710	5583#		
TS265	017772	5596	5611#	
TS266	020124	5653#		
TS267	020162	5672#		
TS27	003436	1199	1206#	
TS270	020204	5679	5699#	
TS271	020376	5749#		
TS272	020432	5761	5767#	
TS273	020466	5779	5785#	
TS274	020556	5806	5812#	
TS275	020646	5834	5848#	
TS276	020664	5852	5858#	
TS277	020706	5862	5868#	
TS3	001736	790#		
TS30	003470	1215	1221#	
TS300	020736	5873	5879#	
TS301	020762	5883	5889#	
TS302	021004	5893	5903#	
TS303	021022	5908	5918#	
TS304	021052	5924	5930#	
TS305	021114	5938	5944#	
TS306	021344	6039#		
TS307	021544	6095	6100#	
TS31	003516	1229	1236#	
TS310	022004	6143	6148#	
TS311	022130	6178	6185#	

TS312	022162	6198#			
TS313	022206	6203	6209#		
TS314	022312	6235#			
TS315	022346	6245#			
TS316	022402	6256#			
TS317	022436	6267#			
TS32	003550	1245	1252#		
TS320	022472	6277#			
TS321	022526	6287#			
TS322	022572	6295	6300#		
TS323	022626	6311#			
TS324	022714	6333#			
TS325	022744	6345#			
TS326	023004	6354	6359#		
TS327	023042	6367	6374#		
TS33	003576	1260	1267#		
TS330	023076	6383	6390#		
TS331	023152	6411#			
TS332	023212	6422	6428#		
TS333	023276	6448#			
TS334	023346	6474#			
TS335	023606	6540#			
TS336	023730	6565#			
TS337	024040	6586#			
TS34	003630	1276	1283#		
TS340	024172	6610#			
TS341	024236	6545	6614	6620	6626#
TS342	024324	6630	6649#		
TS343	024500	6682#			
TS344	024532	6692	6701#		
TS345	024562	6708	6719#		
TS346	024612	6725	6739#		
TS347	024730	6772#			
TS35	003656	1291	1298#		
TS350	024766	6785#			
TS351	037456	9318#			
TS352	037512	9337#			
TS353	037554	9356#			
TS354	037642	9375	9384#		
TS355	037700	9391	9414#		
TS356	037724	9426#			
TS357	037744	9438#			
TS36	003710	1307	1315#		
TS360	037764	9450#			
TS361	040004	9462#			
TS362	040024	9474#			
TS363	040142	9520#			
TS364	040176	9537#			
TS365	040260	9556	9566#		
TS366	040350	9593	9600#		
TS367	040446	9629	9638#		
TS37	003736	1323	1330#		
TS370	040552	9668	9675#		
TS371	040656	9704	9711#		
TS372	041022	9746#			
TS373	041206	9752	9829#		

TS374	041662	9952	9964#	
TS375	042212	10068#		
TS376	042560	10131	10142#	
TS377	042734	10192#		
TS4	002132	836#		
TS40	003770	1339	1346#	
TS400	043122	10243#		
TS401	043244	10289#		
TS402	043522	10308	10354#	
TS403	043712	10436#		
TS404	044212	10459	10492	10532#
TS405	044476	10553	10586	10627#
TS406	044652	10664#		
TS407	045244	10669	10748#	
TS41	004016	1354	1361#	
TS410	045402	10779#		
TS411	045446	10790	10800#	
TS412	045600	10837#		
TS413	046104	10904#		
TS414	046600	11041#		
TS415	047332	11102	11183#	
TS416	047732	11303#		
TS417	050014	11334#		
TS42	004050	1370	1391#	
TS420	050556	11556#		
TS421	050650	11592#		
TS422	050704	11617#		
TS423	051134	11705#		
TS424	051302	11762#		
TS425	051350	11790#		
TS426	051710	11903#		
TS427	052212	12007#		
TS43	004072	1395	1401#	
TS430	052516	12112#		
TS431	054326	12509#		
TS432	055214	12707#		
TS433	055522	12808#		
TS434	055666	12873#		
TS435	056032	12935#		
TS436	056230	13008#		
TS437	056366	13069#		
TS44	004112	1404	1410#	
TS440	056542	13135#		
TS441	056716	13202#		
TS442	057040	13254#		
TS443	057172	13313#		
TS444	057760	13458#		
TS445	060334	13561#		
TS446	060730	13671#		
TS447	061160	13738#		
TS45	004132	1413	1419#	
TS450	061372	13804#		
TS451	061676	13904#		
TS452	062506	14080#		
TS453	063306	14252#		
TS454	064206	14444#		

TS455	064444	14519#	
TS456	064670	14606#	
TS457	065444	14779#	
TS46	004152	1422	1458#
TS460	066446	15005#	
TS461	067310	15210#	
TS462	070114	15454#	
TS463	070370	15552#	
TS464	071122	15731#	
TS465	071614	15886#	
TS466	072346	16064#	
TS467	072700	16185#	
TS47	004200	1468	1484#
TS470	073202	16318#	
TS471	073564	16463#	
TS472	074116	16601#	
TS473	074530	16752#	
TS474	075464	16977#	
TS475	076700	17243#	
TS476	077340	17403#	
TS477	100044	17565#	
TS5	002254	865#	
TS50	004236	1498	1511#
TS500	100742	17732#	
TS501	101036	17750	17779#
TS502	101220	17839#	
TS503	101506	17923#	
TS504	101660	17985#	
TS505	102044	18045#	
TS506	102210	18098#	
TS507	102354	18148#	
TS51	004256	1518	1533#
TS510	102512	18194#	
TS511	102666	18245#	
TS512	103320	18425#	
TS513	103752	18600#	
TS514	104012	18627#	
TS515	104132	18674#	
TS516	104172	18699#	
TS517	104232	18726#	
TS52	004302	1542	1556#
TS520	104366	18782#	
TS521	104524	18833#	
TS522	104652	18880#	
TS523	105010	18931#	
TS524	105156	18982#	
TS525	105324	19034#	
TS526	105456	19080#	
TS527	105626	19131#	
TS53	004336	1569	1586#
TS530	105744	19174#	
TS531	106102	19223#	
TS532	106226	19271#	
TS533	106356	19318#	
TS534	106506	19366#	
TS535	106650	19416#	

TS536	107010	19467#	
TS537	107152	19516#	
TS54	004376	1601	1618#
TS540	107302	19563#	
TS541	107436	19609#	
TS542	107600	19661#	
TS543	107752	19714#	
TS544	110672	19982#	
TS545	110754	20018#	
TS546	111030	20055#	
TS547	111130	20086#	
TS55	004440	1634	1651#
TS550	111236	20119#	
TS551	111336	20151#	
TS552	111432	20183#	
TS553	111542	20220#	
TS554	111606	20249#	
TS555	111704	20288#	
TS556	112704	20532#	
TS557	113460	20705#	
TS56	004502	1667	1679#
TS560	114764	21008#	
TS561	115076	21047#	
TS562	115210	21087#	
TS563	115322	21125#	
TS564	115444	21165#	
TS565	115570	21205#	
TS566	115714	21246#	
TS567	116052	21287#	
TS57	004552	1698	1710#
TS570	116130	21315#	
TS571	116206	21344#	
TS572	117306	21604#	
TS573	117346	21628#	
TS574	117704	21759#	
TS575	120012	21800#	
TS576	120102	21830#	
TS577	120152	21847#	
TS6	002402	902#	
TS60	004624	1734	1739#
TS600	120610	21973#	
TS601	120636	21987#	
TS602	120664	22000#	
TS603	120752	22027#	
TS604	121004	22035	22043#
TS605	121032	22056#	
TS606	121060	22073#	
TS607	121154	22088	22095#
TS61	004704	1763	1768#
TS610	121272	22123#	
TS611	121410	22152#	
TS612	121502	22176#	
TS613	121602	22201#	
TS614	121720	22229#	
TS615	122026	22255#	
TS616	122076	22278#	

TS617	122140	22286	22294#	
TS62	004756	1788	1808#	
TS620	122272	22322#		
TS621	122424	22351#		
TS622	122564	22382#		
TS623	122666	22406#		
TS624	122770	22429#		
TS625	123064	22453#		
TS626	123250	22495#		
TS627	123276	22507#		
TS63	005012	1821	1840#	
TS630	123432	22538#		
TS631	123534	22543	22567#	
TS632	123722	22611#		
TS633	124056	22615	22645#	
TS634	124162	22649	22671#	
TS635	124260	22694#		
TS636	124532	22765#		
TS637	124560	22779#		
TS64	005062	1859	1879#	
TS640	124606	22792#		
TS641	124670	22796	22819#	
TS642	124740	22823	22832	22840#
TS643	124766	22853#		
TS644	125014	22870#		
TS645	125126	22886	22891	22898#
TS646	125262	22930#		
TS647	125416	22963#		
TS65	005126	1896	1902#	
TS650	125510	22987#		
TS651	125610	23012#		
TS652	125726	23040#		
TS653	126052	23044	23071#	
TS654	126136	23098#		
TS655	126164	23104	23112#	
TS656	126274	23137#		
TS657	126376	23163#		
TS66	005230	1931	1945#	
TS660	126522	23192#		
TS661	126610	23214#		
TS662	126722	23218	23241#	
TS663	127002	23263#		
TS664	127076	23267	23291#	
TS665	127154	23308#		
TS666	127356	23366#		
TS667	127504	23370	23406#	
TS67	005266	1959	1981#	
TS7	002524	937#		
TS70	005334	1998	2013#	
TS71	005370	2023	2039#	
TS72	005426	2050	2057#	
TS73	005464	2072	2077#	
TS74	005534	2097	2102#	
TS75	005574	2117	2122#	
TS76	005640	2138	2152#	
TS77	005674	2161	2177#	

```
TTBDON   107276           19540    19554#
TTBTP1   107254           19517    19522    19530    19536    19543#
TTBTP2   107264           19518    19531    19547#
TTB10    107274           19534    19537    19539    19552#
TTB2     107210           19527#
TTCDON   116046           21268    21277#
TTCSR =  177564             533#    6531     6552*    6558*    6570*    6582*    6589*    6605*    6655*    6660     6663     6666*    6678*
                           6786*   21859    21862*   21871*   21873    21876*   21894*   23595    23599
TTCTB0   116024           21248    21271#
TTCTB1   116030           21257    21264    21266    21272#
TTCTB2   116040           21256    21262    21274#
TTC10    116044           21253    21261    21263    21265    21267    21275#
TTC2     115762           21259#
TTTA1    102166           18078#
TTTA2    102170           18052*   18058    18062    18079#
TTTBFO   102154           18047    18052    18064    18073#   18079
TTTDON   102204           18070    18088#
TTTTP1   102172           18055    18065    18080#
TTT10    102202           18057    18063    18068    18085#
TTT2     102120           18060#
TTT3     102144           18067#   18069
TVECT    120546           21955#   22096    22097*   22117*   22154    22155*   22162*   22170*   22179    22180*   22189*   22195*   22203
                          22205*   22216*   22223*   22232    22233*   22245*   23416    23424*   23520*
TVECT2   124476           22752#   22899    22900*   22924*   22965    22966*   22973*   22981*   22990    22991*   23000*   23006*   23014
                          23016*   23027*   23034*   23048    23049*   23061*   23414    23428*   23518*
TYPE     131640           23557    23589    23593#
T105     052312           12033    12035#
T12      052314           12039#
T13      052330           12042#   12043
T14      052356           12053#
T15      052360           12054#
T16      052362           12055#
T17      052400           12060    12062#
T2       052230           12012#   12013
T20      052440           12073#   12076
T23      052446           12074    12076#
T3       052254           12022#
T4       052256           12023#
T5       052260           12024#
T6       052304           12032#   12035
UDONE    057760           13419    13452#
UERRO    057314           13322    13338#
UERR3    057646           13400    13412#
UFLAG    057744           13314*   13388    13395*   13446#
UIPAR0=  177640            9244#    9452     9594     9669     9736     9791    10075    10752*   10769*   10921*   11419    11475
UIPAR1=  177642            9245#   10922*
UIPAR2=  177644            9246#   10923*
UIPAR3=  177646            9247#   10196    10294*   10839*   10924*
UIPAR4=  177650            9248#   10092*   10113*   10128*   10295*   10840*   10929*   11045*   11193*
UIPAR5=  177652            9249#   10093*   10114*   10129*
UIPAR6=  177654            9250#
UIPAR7=  177656            9251#    9592     9667    10925*
UIPDR0=  177600            9233#    9464     9630     9705     9729     9779    10082    10200    10203    10216    10918    11187    11415
                          11462
UIPDR1=  177602            9234#
UIPDR2=  177604            9235#
```

```
UIPDR3= 177606      9236#   10297*  10841*  10891*
UIPDR4= 177610      9237#   10090*  10115*  10304*  10363*  10842*  11043*  11200*
UIPDR5= 177612      9238#   10091*  10116*
UIPDR6= 177614      9239#
UIPDR7= 177616      9240#    9628    9703
UNIQUE  127326     22757   23324   23357#
UPAT00  057674     13315   13325   13343   13360   13375   13391   13403   13421#
UPAT01  057676     13422#
UPAT02  057700     13423#
UPAT03  057702     13424#
UPAT10  057704     13331   13426#
UPAT11  057706     13427#
UPAT12  057710     13428#
UPAT13  057712     13429#
UPAT20  057714     13349   13431#
UPAT21  057716     13432#
UPAT22  057720     13433#
UPAT23  057722     13434#
UPAT30  057724     13365   13436#
UPAT31  057726     13437#
UPAT32  057730     13438#
UPAT33  057732     13439#
UPAT40  057734     13380   13405   13441#
UPAT41  057736     13442#
UPAT42  057740     13443#
UPAT43  057742     13444#
UROM1   057752     13327*  13345*  13362*  13377*  13449#
UROM2   057754     13328*  13346*  13363*  13378*  13450#
UROM3   057756     13329*  13347*  13364*  13379*  13451#
USESTK= 000600       546#   5568    5575    5587    5934    5937    5946    9388   10756   10771   10859   10942   11061
                   11203   11310   11328
USP1    017472      5518#   5519
USP1A   017510      5520    5523#
USP2    017546      5540    5543#
USP3    017564      5545    5547#
USP4    017612      5551    5553#
USRM  = 140000       517#   5515    5521    5523    5537    5541    5543    5548    5550    5554    5567    5572    5574
                    5586    5590    5592
UTMP1   057746     13320*  13327   13377   13396*  13447#
UTMP2   057750     13321*  13345   13362   13397*  13448#
UUBDON  107432     19590   19600#
UUBTP1  107346     19565   19569   19570*  19576#  19580
UUBTP2  107420     19564   19581   19593#
UUB10   107430     19584   19587   19589   19598#
UUB2    107344     19575#
UUCBF0  116116     21292   21296   21303#
UUCDON  116124     21297   21305#
UUCTP1  116104     21290   21300#
UUC2    116072     21294#
UUUA1   102332     18105*  18129#
UUUA2   102334     18111   18113   18130#
UUUBF0  102320     18100   18105   18115   18124#  18129
UUUDON  102350     18121   18138#
UUUTP1  102336     18108   18116   18131#
UUU10   102346     18110   18114   18119   18136#
UUU2    102264     18112#
```

```
UUU3     102310      18118#  18120
U0       057206      13317#  13318
U1       057230      13322#  13398
U10      057520      13381#
U11      057522      13382#
U12      057540      13386   13388#
U13      057556      13393#  13394
U14      057610      13389   13400#
U15      057636      13406#
U16      057640      13407#  13412
U2       057300      13332#
U3       057316      13337   13340#
U4       057360      13351#
U5       057376      13356   13358#
U6       057440      13367#
U7       057456      13371   13373#
VDEC     022340       6238    6241#
VDEC10   022514       6279    6282#
VDEC11   022550       6289    6292#
VDEC12   022554       6290    6293#
VDEC13   022614       6302    6305#
VDEC14   022620       6303    6306#
VDEC2    022334       6237    6240#
VDEC3    022374       6248    6251#
VDEC4    022370       6247    6250#
VDEC5    022430       6259    6262#
VDEC6    022424       6258    6261#
VDEC7    022464       6270    6273#
VDEC8    022460       6269    6272#
VDEC9    022520       6280    6283#
VVBDON   104226      18712   18716#
VVB10    104224      18709   18711   18713#
VVB2     104200      18703#
VVCBF0   116174      21321   21325   21332#
VVCDON   116202      21326   21334#
VVCTP1   116162      21319   21329#
VVC2     116150      21323#
VVVBF0   102464      18152   18160   18164   18166   18173#
VVVDON   102506      18172   18184#
VVVTP1   102474      18158   18167   18177#
VVV10    102504      18157   18165   18170   18182#
VVV2     102426      18162#
WAITER   130214      23490#
WAITIO   130004      23445#  23446
WASR6    037346       9288#
WASSR0   037354       9291#  10320*  10322   10378*  10380   10465*  10468   10559*  10562   10699*  10701   10716*  10717
                     10812*  10815   10853*  10872
WASSR2   037356       9292#   9485*   9487    9505*   9506   10321*  10329   10379*  10387   10466*  10475   10560*  10569
                     10641*  10642   10674*  10676   10682*  10684   10700*  10704   10720*  10721   10731*  10733   10813*
                     10822   10854*  10875
WATE     024442       6658    6670#
WATE1    024374       6660#   6661
WATE2    024410       6663#   6664
WATE3    024436       6668#   6673
WBIT   = 000100       9279#  10159   10167   10178   10209   10217   10228
WDAP00   060320      13545#
```

```
WDAT01   060322     13546#
WDAT02   060324     13547#
WDAT03   060326     13548#
WDONE    060330     13537   13550#
WDONE1   122042     22259#  22262
WDONE2   126060     23074#  23077
WPAT00   060310     13461   13463   13467   13469   13480   13484   13488   13490   13501   13503   13507   13509   13520
                    13524   13528   13530   13540#
WPAT01   060312     13541#
WPAT02   060314     13542#
WPAT03   060316     13543#
WWBDON   110666     19886   19970#
WWB1     107752     19716#
WWB10    110364     19849#
WWB11    110432     19868#
WWB2     110020     19735#
WWB3     110066     19754#
WWB4     110134     19773#
WWB5     110202     19792#
WWB6     110250     19811#
WWB7     110316     19830#
WWCDON   117302     21520   21593#
WWC1     116206     21347#
WWC10    116512     21418#
WWC11    116546     21429#
WWC12    116602     21439#
WWC13    116636     21449#
WWC14    116672     21459#
WWC15    116726     21469#
WWC16    116762     21479#
WWC17    117016     21489#
WWC2     116242     21357#
WWC20    117052     21500#
WWC21    117106     21511#
WWC3     116276     21367#
WWC4     116332     21377#
WWC5     116366     21388#
WWC6     116422     21398#
WWC7     116456     21408#
WWWBF0   102630     18198   18208   18213   18220#
WWWBF1   102650     18206   18208*  18211   18228#
WWWDON   102662     18219   18235#
WWWTP1   102640     18204   18214   18224#
WWW10    102660     18203   18212   18217   18233#
WWW2     102572     18209#
W10      060122     13493   13495#
W11      060134     13497   13499#
W12      060154     13504#
W13      060200     13511#  13514
W14      060206     13512   13514#
W15      060220     13516   13518#
W16      060244     13525#
W17      060270     13532#  13535
W2       060000     13464#
W20      060276     13533   13535#
W3       060024     13471#  13474
```

```
W4       060032      13472   13474#
W5       060044      13476   13478#
W6       060070      13485#
W7       060114      13492#  13495
X        120364      21890#
XAPT11   060706      13653#
XBJF     100626      17608   17610   17622   17624   17637   17639   17685   17687   17700#
XDAT00   060664      13570   13590   13610   13630   13642#
XDAT01   060666      13643#
XDAT02   060670      13644#
XDAT03   060672      13645#
XDONE    060724      13640   13661#
XMIT1    130020      23424   23450#
XMIT2    130102      23428   23467#
XMTCT1   130372      23419*  23450*  23453   23496   23501   23525#
XMTCT2   130374      23420*  23467*  23470   23507   23512   23526#
XOR1     016146       5118    5119    5120    5121    5124#
XOR2     016150       5123    5126#
XOR3     016176       5130    5131    5132    5133    5136#
XPATO    100726      17645*  17646   17652*  17653   17716#
XPATO    100636      17578   17580   17588   17590   17601   17630   17659   17702#
XPAT00   060674      13564   13572   13604   13612   13647#
XPAT01   060676      13648#
XPAT02   060700      13649#
XPAT03   060702      13650#
XPAT1    100646      17704#
XPAT10   060704      13566   13586   13606   13626   13652#
XPAT12   060710      13654#
XPAT13   060712      13655#
XPAT2    100656      17684   17706#
XPAT20   060714      13584   13592   13624   13632   13656#
XPAT21   060716      13657#
XPAT22   060720      13658#
XPAT23   060722      13659#
XPAT3    100666      17602   17616   17631   17708#
XPAT4    100676      17644   17651   17660   17667   17668   17675   17676   17683   17710#
XPAT5    100706      17692   17712#
XPAT6    100716      17693   17714#
XTDONE   100736      17696   17719#
XT1      100044      17566#
XT1A     100060      17566   17569#
XT10     100470      17663   17666#
XT11     100516      17671   17674#
XT12     100544      17679   17682#
XT13     100600      17688   17691#
XT2      100062      17568   17572#
XT2A     100102      17576   17578#
XT2B     100120      17581   17583#
XT3      100132      17585   17587#
XT3A     100154      17591   17593#
XT4      100164      17595   17598#
XT4A     100214      17599   17604#
XT4B     100226      17606   17608#
XT5      100244      17611   17614#
XT5A     100264      17614   17618#
XT5B     100276      17620   17622#
```

XT6	100314	17625	17628#				
XT6A	100340	17628	17633#				
XT6B	100352	17635	17637#				
XT7	100370	17640	17643#				
XT8	100416	17647	17650#				
XT9	100444	17654	17658#				
XXBDON	107574	19636	19651#				
XXBTP1	107552	19610	19615	19616*	19626	19632	19639#
XXBTP2	107562	19611	19627	19643#			
XXB10	107572	19625	19630	19633	19635	19649#	
XXB2	107500	19621#					
XXCDON	117342	21617#					
XXXDON	103314	18341	18415#				
XXX1	102666	18248#					
XXX2	102732	18267#					
XXX3	102776	18286#					
XXX4	103042	18305#					
XXX5	103106	18324#					
X10	060474	13595	13597#				
X11	060510	13600	13602#				
X12	060530	13607#					
X13	060554	13614#	13617				
X14	060562	13615	13617#				
X15	060576	13620	13622#				
X16	060616	13627#					
X17	060642	13634#	13637				
X2	060354	13567#					
X20	060650	13635	13637#				
X3	060400	13574#	13577				
X4	060406	13575	13577#				
X5	060422	13580	13582#				
X6	060442	13587#					
X7	060466	13594#	13597				
Y	120366	21867	21891#				
YBR	001534	725*	728*	738#			
YDAT00	061114	13684	13709#				
YDAT01	061116	13710#					
YDAT02	061120	13711#					
YDAT03	061122	13712#					
YDONE	061154	13702	13729#				
YFLAG	061104	13672*	13695	13697*	13704#		
YNTAB	021220	712	5973#				
YPAT00	061124	13673	13699	13714#			
YPAT01	061126	13715#					
YPAT02	061130	13716#					
YPAT03	061132	13717#					
YPAT10	061134	13674	13698	13719#			
YPAT11	061136	13720#					
YPAT12	061140	13721#					
YPAT13	061142	13722#					
YPAT20	061144	13678	13724#				
YPAT21	061146	13725#					
YPAT22	061150	13726#					
YPAT23	061152	13727#					
YTMP1	061106	13673*	13680	13698*	13705#		
YTMP2	061110	13674*	13687	13699*	13706#		

YTMP3	061112	13675*	13692	13700*	13707#		
YYBDON	107746	19690	19705#				
YYBTP1	107722	19663	19669	19670*	19680	19693#	19701
YYBTP2	107732	19664	19681	19697#			
YYBTP3	107742	19668	19686	19701#			
YYB10	107744	19679	19684	19687	19689	19702#	
YYB2	107650	19675#					
YYCDON	117700	21687	21749#				
YYC1	117346	21631#					
YYC2	117374	21640#					
YYC3	117422	21649#					
YYC4	117450	21659#					
YYC5	117476	21669#					
YYC6	117524	21679#					
YYYDON	103746	18521	18591#				
YYY1	103320	18428#					
YYY2	103364	18447#					
YYY3	103430	18466#					
YYY4	103474	18485#					
YYY5	103540	18504#					
Y1	060756	13676#	13701				
Y2	060776	13681#					
Y3	061022	13688#	13691				
Y4	061042	13693	13695#				
Y5	061102	13696	13702#				
Z	120370	21868*	21886*	21887	21892#		
ZDAT00	061326	13750	13773#				
ZDAT01	061330	13774#					
ZDAT02	061332	13775#					
ZDAT03	061334	13776#					
ZDONE	061366	13767	13793#				
ZFLAG	061320	13739*	13761	13763*	13769#		
ZPAT00	061336	13740	13778#				
ZPAT01	061340	13779#					
ZPAT02	061342	13780#					
ZPAT03	061344	13781#					
ZPAT10	061346	13764	13783#				
ZPAT11	061350	13784#					
ZPAT12	061352	13785#					
ZPAT13	061354	13786#					
ZPAT20	061356	13744	13788#				
ZPAT21	061360	13789#					
ZPAT22	061362	13790#					
ZPAT23	061364	13791#					
ZTMP1	061322	13740*	13746	13753	13764*	13770#	
ZTMP2	061324	13741*	13758	13765*	13771#		
ZZCBF	117774	21766	21772	21785#			
ZZCDON	120006	21779	21789#				
ZZC10	117770	21774	21776	21778	21780#		
ZZC2	117712	21764#	21775				
ZZC3	117732	21769#					
ZZZDON	104006	18612	18617#				
ZZZ10	104004	18609	18611	18614#			
ZZZ2	103760	18604#					
Z1	061200	13742#	13766				
Z2	061220	13747#					

```
Z3        061244        13754#  13757
Z4        061252        13755   13757#
Z5        061264        13759   13761#
Z6        061316        13762   13767#
$APTHD    001030          597     603#
$BDADR    127224        23326#
$BDDAT    127226        23327#
$CPUOP    001026          579#
$DEVCT    001010          570#
$DOAGN    131570        23552   23571#
$ENDAD    131522          556   23561#
$ENV      001020          575#    675    5494    6032    6541    6611    6627    6650    9749   10666   11498   22001   22523
                        22540   22612   22646   22695   22793   22820   22883   22914   22946   23041   23080   23215   23264
                        23367   23407
$ENVM     001021          576#  23593
$EOPCT    131456        23551#  23555*
$ETABL    001020          574#
$ETEND    001030          586#    609
$FATAL    001002          567#    673*    828*    830*  11518*  11520*  21912*  21914*  22478*  22480*  22738*  22740*  23339*
                        23341*  23538*  23540*
$GDADR    127230        23328#
$GDDAT    127232        23329#
$HIBTS    001030          604#
$MAIL     001000          565#    605     609
$MBADR    001032          605#
$MSGAD    001014          572#
$MSGLG    001016          573#
$MSGTY    001000          566#    674*    831*  11521*  21915*  22481*  22741*  23342*  23541*
$PASS     001006          569#    662*   5496    6543    6613    6629    6652    6962    6992    7020    7048    7077    7107
                         7400    7437    7474    9751   10668   11500   22003   22525   22542   22614   22648   22697   22795
                        22822   22885   22916   22948   23043   23082   23217   23266   23369   23409   23553*  23554*
$PASTM    001036          607#
$SETUP= 000020           6929#
$STUP  = 177777          6929#
$SVPC  = 000400           554#    559
$SWR   = 000000           507#
$SWREG    001022          577#   6034
$TESTN    001004          568#    656     672*   5642    6937    9311*  11550*  21966*  22488*  22758*  23360*
$TMPO     037362         9294#   9849*   9855    9873*   9879    9897*   9903    9973*   9979    9997*  10003   10021*  10027
                        10045*  10051   10094*  10306*  10336   10365*  10394   10694*  10701   10861*  10867*  10871*  10874*
                        10878*  10879   21853*  21895   23414*  23518   23565*  23570
$TMP1     037364         9295#  10691*  10703*  10706*  10707   10713*  10719*  10723*  10724   21854*  21896   23415*  23519
$TMP2     037366         9296#  10695*  10704   10998*  10999   11023*  11024   11128   11129*  11161*  11163   11261*  11262
                        11286*  11287   17738*  17898*  19988*  20060*  23416*  23520
$TMP3     037370         9297#  23417*  23521
$TMP4     037372         9298#  23418*  23522
$TN    = 000670           507#    707     711#    758     762#    787     791#    833     837#    862     866#    899     903#
                          934     938#    963     967#   1013    1017#   1020    1023    1027#   1030    1033    1037#   1040
                         1043    1047#   1049    1066    1070#   1074    1077    1081#   1087    1090    1094#   1101    1105
                         1109#   1113    1116    1120#   1127    1151    1155#   1158    1161    1165#   1167    1170    1174#
                         1176    1179    1183#   1185    1188    1192#   1199    1203    1207#   1215    1218    1222#   1229
                         1233    1237#   1245    1249    1253#   1260    1264    1268#   1276    1280    1284#   1291    1295
                         1299#   1307    1312    1316#   1323    1327    1331#   1339    1343    1347#   1354    1358    1362#
                         1370    1388    1392#   1395    1398    1402#   1404    1407    1411#   1413    1416    1420#   1422
                         1455    1459#   1468    1481    1485#   1498    1508    1512#   1518    1530    1534#   1542    1553
                         1557#   1569    1583    1587#   1601    1615    1619#   1634    1648    1652#   1667    1676    1680#
```

```
     1698    1707    1711#   1734    1736    1740#   1763    1765    1769#   1788    1805    1809#   1821
     1837    1841#   1859    1876    1880#   1896    1899    1903#   1931    1942    1946#   1959    1978
     1982#   1998    2010    2014#   2023    2036    2040#   2050    2054    2058#   2072    2074    2078#
     2097    2099    2103#   2117    2119    2123#   2138    2149    2153#   2161    2174    2178#   2185
     2197    2201#   2208    2220    2224#   2232    2243    2247#   2266    2277    2281#   2294    2305
     2309#   2335    2345    2349#   2364    2376    2380#   2405    2415    2419#   2431    2441    2445#
     2460    2470    2474#   2487    2497    2501#   2514    2523    2527#   2532    2539    2543#   2548
     2555    2559#   2570    2581    2585#   2613    2621    2625#   2639    2647    2651#   2672    2679
     2683#   2697    2707    2711#   2717    2726    2730#   2736    2746    2750#   2756    2770    2774#
     2786    2796    2800#   2807    2817    2821#   2830    2842    2846#   2854    2863    2867#   2878
     2888    2892#   2896    2905    2909#   2913    2922    2926#   2930    2933    2937#   2948    2951
     2955#   2966    2969    2973#   2988    2991    2995#   3021    3025    3029#   3056    3059    3063#
     3079    3082    3086#   3110    3113    3117#   3133    3136    3140#   3155    3158    3162#   3177
     3186    3190#   3201    3210    3214#   3230    3238    3242#   3263    3271    3275#   3296    3306
     3310#   3323    3334    3338#   3358    3370    3374#   3395    3406    3410#   3431    3441    3445#
     3465    3480    3484#   3490    3509    3513#   3519    3537    3541#   3547    3559    3563#   3569
     3579    3583#   3598    3611    3615#   3642    3653    3657#   3690    3701    3705#   3726    3738
     3742#   3750    3764    3768#   3777    3789    3793#   3798    3809    3813#   3819    3832    3836#
     3841    3851    3855#   3859    3870    3874#   3881    3892    3896#   3899    3910    3914#   3921
     3934    3938#   3945    3961    3965#   3969    3985    3989#   3993    4031    4035#   4089    4109
     4113#   4206    4218    4222#   4230    4246    4250#   4264    4267    4271#   4286    4289    4293#
     4307    4310    4314#   4330    4345    4349#   4374    4378    4382#   4414    4428    4432#   4438
     4442    4446#   4461    4464    4468#   4483    4497    4501#   4537    4541    4545#   4569    4583
     4587#   4614    4618    4622#   4655    4659    4663#   4669    4684    4688#   4721    4725    4729#
     4763    4777    4781#   4815    4818    4822#   4855    4858    4862#   4897    4900    4904#   4942
     4947    4951#   4998    5002    5006#   5022    5027    5031#   5060    5071    5075#   5098    5109
     5113#   5135    5145    5149#   5164    5174    5178#   5205    5218    5222#   5235    5239    5243#
     5251    5254    5258#   5267    5270    5274#   5282    5285    5289#   5297    5300    5304#   5312
     5315    5319#   5327    5338    5342#   5353    5356    5360#   5374    5377    5381#   5395    5398
     5402#   5416    5419    5423#   5437    5440    5444#   5458    5461    5465#   5479    5490    5494#
     5497    5502    5511    5515#   5533    5537#   5562    5566#   5580    5584#   5596    5608    5612#
     5650    5654#   5669    5673#   5679    5696    5700#   5746    5750#   5761    5764    5768#   5779
     5782    5786#   5806    5809    5813#   5834    5845    5849#   5852    5855    5859#   5862    5865
     5869#   5873    5876    5880#   5883    5886    5890#   5893    5900    5904#   5908    5915    5919#
     5924    5927    5931#   5938    5941    5945#   6036    6040#   6095    6097    6101#   6143    6145
     6149#   6178    6182    6186#   6195    6199#   6203    6206    6210#   6232    6236#   6242    6246#
     6253    6257#   6264    6268#   6274    6278#   6284    6288#   6295    6297    6301#   6308    6312#
     6330    6334#   6342    6346#   6354    6356    6360#   6367    6371    6375#   6383    6387    6391#
1    6408    6412#   6422    6425    6429#   6445    6449#   6471    6475#   6537    6541#   6562    6566#
     6583    6587#   6607    6611#   6614    6620    6623    6627#   6630    6646    6650#   6679    6683#
     6692    6698    6702#   6708    6716    6720#   6725    6736    6740#   6769    6773#   6782    6786#
     9315    9319#   9334    9338#   9353    9357#   9375    9381    9385#   9391    9411    9415#   9423
     9427#   9435    9439#   9447    9451#   9459    9463#   9471    9475#   9517    9521#   9534    9538#
     9556    9563    9567#   9593    9597    9601#   9629    9635    9639#   9668    9672    9676#   9704
     9708    9712#   9743    9747#   9752    9826    9830#   9952    9961    9965#   10065   10069#  10131
     10139   10143#  10189   10193#  10240   10244#  10286   10290#  10308   10351   10355#  10433   10437#
     10459   10492   10529   10533#  10553   10586   10624   10628#  10661   10665#  10669   10745   10749#
     10776   10780#  10790   10797   10801#  10834   10838#  10901   10905#  11038   11042#  11102   11180
     11184#  11300   11304#  11331   11335#  11553   11557#  11589   11593#  11614   11618#  11702   11706#
     11759   11763#  11787   11791#  11900   11904#  12004   12008#  12109   12113#  12506   12510#  12704
     12708#  12805   12809#  12870   12874#  12932   12936#  13005   13009#  13066   13070#  13132   13136#
     13199   13203#  13251   13255#  13310   13314#  13455   13459#  13558   13562#  13668   13672#  13735
     13739#  13801   13805#  13901   13905#  14077   14081#  14249   14253#  14441   14445#  14516   14520#
     14603   14607#  14776   14780#  15002   15006#  15207   15211#  15451   15455#  15549   15553#  15728
     15732#  15883   15887#  16061   16065#  16182   16186#  16315   16319#  16460   16464#  16598   16602#
     16749   16753#  16974   16978#  17240   17244#  17400   17404#  17562   17566#  17729   17733#  17750
```

```
                        17776   17780#  17836   17840#  17920   17924#  17982   17986#  18042   18046#  18095   18099#  18145
                        18149#  18191   18195#  18242   18246#  18422   18426#  18597   18601#  18624   18628#  18671   18675#
                        18696   18700#  18723   18727#  18779   18783#  18830   18834#  18877   18881#  18928   18932#  18979
                        18983#  19031   19035#  19077   19081#  19128   19132#  19171   19175#  19220   19224#  19268   19272#
                        19315   19319#  19363   19367#  19413   19417#  19464   19468#  19513   19517#  19560   19564#  19606
                        19610#  19658   19662#  19711   19715#  19979   19983#  20015   20019#  20052   20056#  20083   20087#
                        20116   20120#  20148   20152#  20180   20184#  20217   20221#  20246   20250#  20285   20289#  20529
                        20533#  20702   20706#  21005   21009#  21044   21048#  21084   21088#  21122   21126#  21162   21166#
                        21202   21206#  21243   21247#  21284   21288#  21312   21316#  21341   21345#  21601   21605#  21625
                        21629#  21756   21760#  21797   21801#  21827   21831#  21844   21848#  21970   21974#  21984   21988#
                        21997   22001#  22024   22028#  22035   22040   22044#  22053   22057#  22070   22074#  22088   22092
                        22096#  22120   22124#  22149   22153#  22173   22177#  22198   22202#  22226   22230#  22252   22256#
                        22275   22279#  22286   22291   22295#  22319   22323#  22348   22352#  22379   22383#  22403   22407#
                        22426   22430#  22450   22454#  22492   22496#  22504   22508#  22535   22539#  22543   22564   22568#
                        22608   22612#  22615   22642   22646#  22649   22668   22672#  22691   22695#  22762   22766#  22776
                        22780#  22789   22793#  22796   22816   22820#  22823   22832   22837   22841#  22850   22854#  22867
                        22871#  22886   22891   22895   22899#  22927   22931#  22960   22964#  22984   22988#  23009   23013#
                        23037   23041#  23044   23068   23072#  23095   23099#  23104   23109   23113#  23134   23138#  23160
                        23164#  23189   23193#  23211   23215#  23218   23238   23242#  23260   23264#  23267   23288   23292#
                        23305   23309#  23363   23367#  23370   23403   23407#
$TPB      025462         6926#
$TPS      025464         6927#
$TSTM     001034          606#
$UNIT     001012          571#
$UNITM    001040          608#
$USWR     001024          578#  23555
$X      = 127504          711#    762#    791#    837#    866#    903#    938#    967#   1017#   1027#   1037#   1047#   1070#
                         1081#   1094#   1109#   1120#   1155#   1165#   1174#   1183#   1192#   1207#   1222#   1237#   1253#
                         1268#   1284#   1299#   1316#   1331#   1347#   1362#   1392#   1402#   1411#   1420#   1459#   1485#
                         1512#   1534#   1557#   1587#   1619#   1652#   1680#   1711#   1740#   1769#   1809#   1841#   1880#
                         1903#   1946#   1982#   2014#   2040#   2058#   2078#   2103#   2123#   2153#   2178#   2201#   2224#
                         2247#   2281#   2309#   2349#   2380#   2419#   2445#   2474#   2501#   2527#   2543#   2559#   2585#
                         2625#   2651#   2683#   2711#   2730#   2750#   2774#   2800#   2821#   2846#   2867#   2892#   2909#
                         2926#   2937#   2955#   2973#   2995#   3029#   3063#   3086#   3117#   3140#   3162#   3190#   3214#
                         3242#   3275#   3310#   3338#   3374#   3410#   3445#   3484#   3513#   3541#   3563#   3583#   3615#
                         3657#   3705#   3742#   3768#   3793#   3813#   3836#   3855#   3874#   3896#   3914#   3938#   3965#
                         3989#   4035#   4113#   4222#   4250#   4271#   4293#   4314#   4349#   4382#   4432#   4446#   4468#
                         4501#   4545#   4587#   4622#   4663#   4688#   4729#   4781#   4822#   4862#   4904#   4951#   5006#
                         5031#   5075#   5113#   5149#   5178#   5222#   5243#   5258#   5274#   5289#   5304#   5319#   5342#
                         5360#   5381#   5402#   5423#   5444#   5465#   5494#   5515#   5537#   5566#   5584#   5612#   5654#
                         5673#   5700#   5750#   5768#   5786#   5813#   5849#   5859#   5869#   5880#   5890#   5904#   5919#
                         5931#   5945#   6040#   6101#   6149#   6186#   6199#   6210#   6236#   6246#   6257#   6268#   6278#
                         6288#   6301#   6312#   6334#   6346#   6360#   6375#   6391#   6412#   6429#   6449#   6475#   6541#
                         6566#   6587#   6611#   6627#   6650#   6683#   6702#   6720#   6740#   6773#   6786#   9319#   9338#
                         9357#   9385#   9415#   9427#   9439#   9451#   9463#   9475#   9521#   9538#   9567#   9601#   9639#
                         9676#   9712#   9747#   9830#   9965#  10069#  10143#  10193#  10244#  10290#  10355#  10437#  10533#
                        10628#  10665#  10749#  10780#  10801#  10838#  10905#  11042#  11184#  11304#  11335#  11557#  11593#
                        11618#  11706#  11763#  11791#  11904#  12008#  12113#  12510#  12708#  12809#  12874#  12936#  13009#
                        13070#  13136#  13203#  13255#  13314#  13459#  13562#  13672#  13739#  13805#  13905#  14081#  14253#
                        14445#  14520#  14607#  14780#  15006#  15211#  15455#  15553#  15732#  15887#  16065#  16186#  16319#
                        16464#  16602#  16753#  16978#  17244#  17404#  17566#  17733#  17780#  17840#  17924#  17986#  18046#
                        18099#  18149#  18195#  18246#  18426#  18601#  18628#  18675#  18700#  18727#  18783#  18834#  18881#
                        18932#  18983#  19035#  19081#  19132#  19175#  19224#  19272#  19319#  19367#  19417#  19468#  19517#
                        19564#  19610#  19662#  19715#  19983#  20019#  20056#  20087#  20120#  20152#  20184#  20221#  20250#
                        20289#  20533#  20706#  21009#  21048#  21088#  21126#  21166#  21206#  21247#  21288#  21316#  21345#
                        21605#  21629#  21760#  21801#  21831#  21848#  21974#  21988#  22001#  22028#  22044#  22057#  22074#
```

```
                  22096#  22124#  22153#  22177#  22202#  22230#  22256#  22279#  22295#  22323#  22352#  22383#  22407#
                  22430#  22454#  22496#  22508#  22539#  22568#  22612#  22646#  22672#  22695#  22766#  22780#  22793#
                  22820#  22841#  22854#  22871#  22899#  22931#  22964#  22988#  23013#  23041#  23072#  23099#  23113#
                  23138#  23164#  23193#  23215#  23242#  23264#  23292#  23309#  23367#  23407#
.      = 131741     550#    554     555#    557#    559#    560#    593     594#    596#    598#    613#    622#    627#
                    630#    636#    644#    650#    653#    660#    711     722     745     750     762     791     832
                    837     866     903     938     967    1017    1027    1037    1047    1070    1081    1094    1109
                   1120    1155    1165    1174    1183    1192    1207    1222    1237    1253    1268    1284    1299
                   1316    1331    1347    1362    1392    1402    1411    1420    1459    1485    1512    1534    1557
                   1587    1619    1652    1680    1711    1740    1769    1809    1841    1880    1903    1946    1982
                   2014    2040    2058    2078    2103    2123    2153    2178    2201    2224    2247    2281    2309
                   2349    2380    2419    2445    2474    2501    2527    2543    2559    2585    2625    2651    2683
                   2711    2730    2750    2774    2800    2821    2846    2867    2892    2909    2926    2937    2955
                   2973    2995    3029    3063    3086    3117    3140    3162    3190    3214    3242    3275    3310
                   3338    3374    3410    3445    3484    3513    3541    3563    3583    3615    3657    3705    3742
                   3768    3793    3813    3836    3855    3874    3896    3914    3938    3965    3989    4035    4038
                   4113    4222    4250    4271    4293    4314    4349    4382    4432    4446    4468    4501    4545
                   4587    4622    4663    4688    4729    4781    4822    4862    4904    4951    5006    5031    5075
                   5113    5149    5178    5222    5243    5258    5274    5289    5304    5319    5342    5360    5381
                   5402    5423    5444    5465    5494    5515    5537    5566    5584    5612    5654    5673    5700
                   5750    5768    5786    5813    5849    5859    5869    5880    5890    5904    5919    5931    5945
                   5956    5957    5958    5959    5960    5961    5962    5963    5964    5965    5966    5967    5968
                   5969    5970    6040    6101    6149    6186    6199    6210    6236    6246    6257    6268    6278
                   6288    6301    6312    6334    6337    6346    6349    6360    6363    6366    6375    6391    6412
                   6429    6436    6449    6475    6541    6566    6587    6611    6627    6650    6683    6702    6720
                   6740    6773    6786    6888    6902#   6904#   6906#   6908#   6910#   6912#   9319    9338    9357
                   9385    9415    9427    9439    9451    9463    9475    9521    9538    9567    9601    9639    9676
                   9712    9747    9830    9965   10069   10143   10193   10244   10290   10355   10437   10533   10628
                  10665   10749   10780   10801   10838   10905   11042   11184   11304   11335   11522   11557   11593
                  11618   11706   11763   11791   11904   12008   12113   12510   12708   12809   12874   12936   13009
                  13020#  13070   13136   13203   13255   13314   13459   13562   13672   13739   13805   13905   14081
                  14253   14445   14520   14607   14780   15006   15211   15455   15553   15732   15887   16065   16186
                  16319   16464   16602   16753   16978   17244   17404   17566   17733   17780   17840   17924   17986
                  18046   18099   18149   18195   18246   18426   18601   18628   18675   18700   18727   18783   18834
                  18881   18932   18983   19035   19081   19132   19175   19224   19272   19319   19367   19417   19468
                  19517   19564   19610   19662   19715   19983   20019   20056   20087   20120   20152   20184   20221
                  20250   20289   20533   20706   21009   21048   21088   21126   21166   21206   21247   21288   21316
                  21345   21605   21629   21760   21801   21831   21848   21916   21974   21988   22001   22028   22044
                  22057   22074   22096   22124   22153   22177   22202   22230   22256   22279   22295   22323   22352
                  22383   22407   22430   22454   22482   22496   22508   22539   22568   22612   22646   22672   22695
                  22742   22766   22780   22793   22820   22841   22854   22871   22899   22931   22964   22988   23013
                  23041   23072   23099   23113   23138   23164   23193   23215   23242   23264   23292   23309   23343
                  23367   23407   23542   23544#  23545#
.RSET    120474   11582   11608   11695   11753   11780   11893   11997   12101   12499   12697   12799   12863   12925
                  12998   13059   13125   13192   13244   13304   13551   13662   13730   13794   13896   14071   14244
                  14436   14511   14588   14769   14995   15198   15442   15541   15720   15875   16053   16174   16306
                  16451   16589   16739   16965   17231   17391   17555   17720   17767   17827   17914   17976   18036
                  18089   18139   18185   18236   18416   18592   18618   18665   18690   18717   18773   18824   18872
                  18923   18974   19026   19072   19123   19166   19215   19263   19310   19358   19407   19458   19508
                  19555   19601   19652   19706   19971   20008   20045   20078   20111   20143   20175   20209   20239
                  20277   20522   20695   20997   21037   21076   21114   21154   21195   21236   21278   21306   21335
                  21594   21618   21750   21790   21819   21897   21922#
.$X    = 001030     593#    598
```

ACCMAC	11532#	12114	12140	12166	12192	12218	12244	12270	12296	12322	12348	12374	12400		
ADDTST	9286#	9844	9868	9892	9968	9992	10016	10040							
APTSKP	507#	5494	6611	6627	9749	10665	22539	22612	22646	22793	22820	22882	23041	23215	23264
	23367														
COMMEN	1#														
ENDCOM	1#														
ERROR	507#	739	767	782	796	811	842	857	871	886	908	923	943	958	972
	987	1020	1030	1040	1049	1074	1084	1087	1101	1113	1127	1158	1167	1176	1185
	1199	1215	1229	1245	1260	1276	1291	1307	1323	1339	1354	1370	1395	1404	1413
	1422	1460	1465	1468	1487	1498	1513	1518	1536	1542	1561	1569	1592	1601	1625
	1634	1658	1667	1687	1698	1717	1722	1730	1734	1747	1752	1759	1763	1776	1785
	1788	1814	1821	1848	1859	1886	1896	1914	1918	1923	1926	1931	1950	1959	1991
	1998	2018	2023	2045	2050	2065	2069	2072	2089	2093	2097	2113	2117	2131	2138
	2155	2161	2185	2208	2232	2255	2266	2289	2294	2317	2321	2330	2335	2359	2364
	2391	2400	2405	2427	2431	2455	2460	2483	2487	2510	2514	2532	2548	2566	2570
	2587	2594	2599	2608	2613	2634	2639	2655	2658	2662	2668	2672	2691	2697	2717
	2736	2756	2782	2786	2807	2826	2830	2851	2854	2874	2878	2896	2913	2930	2944
	2948	2962	2966	2979	2984	2988	3003	3007	3013	3018	3021	3040	3044	3051	3056
	3072	3076	3079	3095	3099	3103	3107	3110	3126	3130	3135	3148	3152	3155	3170
	3174	3177	3197	3201	3221	3226	3230	3247	3251	3256	3260	3263	3281	3285	3288
	3292	3296	3316	3320	3323	3343	3346	3351	3355	3358	3380	3384	3387	3392	3395
	3416	3420	3423	3428	3431	3451	3455	3458	3462	3465	3490	3519	3547	3569	3589
	3598	3622	3631	3642	3666	3677	3690	3710	3718	3726	3750	3777	3798	3819	3838
	3841	3859	3878	3881	3899	3918	3921	3942	3945	3969	3993	4039	4042	4050	4056
	4059	4066	4073	4081	4089	4124	4136	4149	4161	4173	4183	4193	4206	4228	4230
	4255	4264	4277	4286	4299	4307	4321	4330	4355	4365	4374	4388	4396	4404	4414
	4438	4452	4461	4474	4483	4506	4514	4522	4528	4537	4551	4560	4569	4594	4604
	4614	4628	4637	4647	4655	4669	4694	4703	4712	4721	4736	4745	4754	4763	4788
	4797	4805	4815	4829	4838	4847	4855	4869	4878	4886	4897	4911	4921	4930	4942
	4956	4962	4968	4972	4984	4988	4994	4998	5012	5016	5022	5036	5042	5047	5056
	5060	5084	5098	5123	5135	5154	5164	5185	5192	5201	5205	5226	5235	5248	5251
	5264	5267	5279	5282	5294	5297	5309	5312	5324	5327	5345	5353	5367	5371	5374
	5388	5392	5395	5409	5413	5416	5430	5434	5437	5451	5455	5458	5472	5476	5479
	5502	5522	5542	5545	5552	5573	5577	5593	5596	5639	5660	5679	5710	5726	5755
	5761	5773	5779	5793	5797	5802	5806	5820	5825	5830	5834	5852	5862	5873	5883
	5893	5908	5924	5938	5953	5997	5999	6001	6003	6005	6007	6009	6011	6013	6043
	6049	6055	6062	6069	6076	6082	6089	6095	6107	6116	6125	6134	6143	6151	6155
	6159	6163	6166	6170	6174	6178	6192	6203	6218	6224	6228	6295	6325	6340	6352
	6354	6367	6383	6401	6405	6419	6422	6439	6441	6456	6485	6490	6493	6511	6518
	6521	6556	6576	6578	6598	6620	6639	6669	6671	6674	6690	6692	6707	6723	6725
	6746	6747	6750	6753	6756	6836	6839	6841	6971	6975	7000	7004	7028	7032	7056
	7060	7085	7089	7115	7119	7199	7214	7217	7231	7234	7248	7251	7265	7268	7282
	7285	7299	7302	7316	7319	7333	7336	7350	7353	7367	7370	7408	7412	7417	7421
	7445	7449	7454	7457	7482	7486	7491	7494	7603	7619	7622	7639	7642	7660	7663
	7681	7684	7701	7704	7722	7725	7743	7746	7763	7766	7769	7791	7794	7797	7814
	7817	7820	7837	7840	7843	7860	7863	7866	7883	7886	7889	7906	7909	7912	7929
	7932	7935	7952	7955	7958	7975	7978	7981	7998	8001	8004	8033	8036	8039	8054
	8057	8060	8075	8078	8081	8097	8100	8103	8119	8122	8125	8140	8143	8146	8162
	8165	8168	8184	8187	8190	8205	8208	8211	8226	8229	8232	8248	8251	8254	8270
	8273	8276	8291	8294	8297	8312	8315	8318	8334	8337	8340	8356	8359	8362	8377
	8380	8383	8399	8402	8405	8425	8428	8431	8446	8449	8452	8467	8470	8473	8488
	8491	8494	8509	8512	8515	8530	8533	8536	8551	8554	8557	8572	8575	8578	8593
	8596	8599	8614	8617	8620	8642	8645	8648	8663	8666	8669	8685	8688	8691	8707
	8710	8713	8728	8731	8734	8750	8753	8756	8772	8775	8778	8793	8796	8799	8814
	8817	8820	8836	8839	8842	8858	8861	8864	8879	8882	8885	8901	8904	8907	8923
	8926	8929	8944	8947	8950	8967	8984	9003	9006	9009	9024	9027	9030	9045	9048

	9051	9066	9069	9072	9087	9090	9093	9108	9111	9114	9129	9132	9135	9150	9153
	9156	9171	9174	9177	9192	9195	9198	9211	9214	9325	9344	9364	9375	9391	9480
	9488	9499	9508	9526	9540	9547	9555	9572	9582	9606	9618	9648	9660	9685	9696
	9760	9772	9783	9796	9858	9882	9906	9931	9952	9982	10006	10030	10054	10100	10133
	10160	10170	10179	10210	10220	10229	10252	10264	10308	10315	10323	10330	10367	10373	10381
	10388	10459	10469	10476	10485	10553	10563	10570	10579	10643	10677	10685	10708	10725	10734
	10763	10790	10807	10816	10823	10880	10944	10957	10968	10978	10989	11003	11015	11029	11055
	11072	11088	11102	11104	11119	11136	11152	11170	11208	11221	11232	11242	11253	11266	11278
	11291	11312	11323	11343	11350	11441	11455	11468	11482	11575	11604	11626	11636	11646	11655
	11665	11674	11683	11692	11720	11734	11739	11751	11776	11812	11815	11821	11827	11845	11848
	11854	11860	11930	11957	11971	12033	12060	12074	12461	12648	12732	12735	12758	12761	12765
	12781	12826	12832	12837	12889	12895	12900	12963	12972	13026	13034	13038	13090	13096	13157
	13164	13221	13225	13274	13279	13337	13356	13371	13386	13409	13413	13419	13472	13476	13493
	13497	13512	13516	13533	13537	13575	13580	13595	13600	13615	13620	13635	13640	13689	13693
	13755	13759	13818	13837	13859	13918	13922	13937	13941	13957	13961	13977	13981	13996	14000
	14015	14019	14094	14099	14114	14118	14135	14139	14155	14159	14174	14178	14193	14197	14266
	14271	14286	14291	14306	14311	14326	14330	14345	14349	14364	14369	14384	14389	14458	14463
	14478	14483	14533	14537	14553	14557	14622	14626	14646	14651	14669	14674	14689	14694	14709
	14714	14795	14800	14812	14816	14825	14830	14836	14853	14858	14872	14881	14886	14892	14909
	14914	14926	14930	14939	14944	14950	15014	15023	15028	15041	15052	15058	15070	15085	15103
	15127	15133	15148	15154	15423	15430	15472	15477	15494	15498	15513	15518	15526	15530	15706
	15709	15712	15863	15868	16039	16042	16045	16162	16167	16292	16298	16437	16443	16556	16559
	16573	16578	16581	16707	16711	16724	16728	16731	16938	16941	16945	16948	16951	17208	17216
	17220	17254	17276	17298	17360	17363	17367	17370	17373	17418	17447	17525	17533	17537	17569
	17576	17581	17585	17591	17595	17603	17606	17611	17617	17620	17625	17632	17635	17640	17647
	17654	17663	17671	17679	17688	17696	17745	17750	17760	17763	17800	17824	17861	17883	17901
	17911	17957	18033	18085	18136	18182	18233	18411	18587	18614	18651	18687	18713	18754	18821
	18869	18920	18971	19023	19069	19120	19163	19212	19260	19307	19355	19404	19455	19505	19552
	19598	19649	19702	19963	20004	20042	20075	20108	20140	20172	20206	20234	20262	20266	20514
	20691	20989	21033	21072	21111	21151	21192	21233	21275	21297	21326	21586	21734	21737	21741
	21780	21809	21812	21866	21882	21977	21991	22013	22019	22035	22047	22060	22075	22079	22083
	22088	22100	22105	22109	22115	22128	22133	22137	22143	22159	22166	22185	22192	22213	22220
	22242	22264	22271	22286	22308	22312	22334	22343	22364	22372	22396	22420	22439	22442	22465
	22499	22512	22517	22521	22531	22546	22553	22559	22578	22587	22597	22628	22634	22662	22683
	22724	22769	22783	22805	22811	22832	22844	22857	22872	22876	22880	22891	22903	22908	22912
	22922	22935	22940	22944	22954	22970	22977	22996	23003	23024	23031	23058	23077	23087	23104
	23121	23127	23146	23153	23174	23182	23204	23232	23249	23252	23278	23282	23298	23301	23316
	23320	23392	23463	23480	23497	23503	23508	23514							
ESCAPE	1#														
GETPRI	1#														
GETSWR	1#														
HMAC1	11528#	12520	12524	12528	12532	12536									
HMAC2	11530#	12620	12624	12628	12632	12636									
HMAC3	11531#	12545	12558	12571	12584	12597									
JNE	507#	720	743	748											
MSG	11553#	11589#	11614#	11702#	11759#	11787#	11900#	12003#	12108#	12506#	12704#	12805#	12869#	12931#	13004#
	13065#	13131#	13198#	13250#	13310#	13455#	13558#	13668#	13735#	13801#	13901#	14077#	14249#	14441#	14516#
	14603#	14775#	15001#	15206#	15450#	15548#	15727#	15882#	16060#	16181#	16314#	16459#	16597#	16748#	16973#
	17239#	17399#	17561#	17728#	17775#	17835#	17920#	17982#	18042#	18095#	18145#	18191#	18242#	18422#	18597#
	18624#	18671#	18695#	19711#	20216#	20245#	20284#	20528#	20701#	21284#	21312#	21341#	21601#	21625#	21756#
MSG1	9315#	18723#	19220#	19978#											
MSG10	9447#														
MSG11	9459#														
MSG12	9471#														
MSG13	9517#														
MSG14	9534#														

```
MSG15     9559#
MSG16     9597#
MSG17     9635#
MSG2      9334#  21004#
MSG20     9672#
MSG21     9708#
MSG21A    9743#
MSG22     9826#
MSG23     9961#
MSG24    10065#
MSG25    10139#
MSG26    10189#
MSG27    10240#
MSG3      9353#
MSG30    10286#
MSG31    10351#
MSG32    10432#
MSG33    10528#
MSG34    10623#
MSG35    10660#
MSG36    10744#
MSG36A   10776#
MSG37    10796#
MSG4      9381#
MSG40    10833#
MSG41    10901#
MSG42    11038#
MSG43    11180#
MSG44    11300#
MSG45    11331#
MSG5      9410#
MSG6      9423#
MSG7      9435#
MULT         1#
NEWTST       1#    507#    707     758     787     833     862     899     934     963    1013    1023    1033    1043    1066
          1077    1090    1105    1116    1151    1161    1170    1179    1188    1203    1218    1233    1249    1264    1280
          1295    1312    1327    1343    1358    1388    1398    1407    1416    1455    1481    1508    1530    1553    1583
          1615    1648    1676    1707    1736    1765    1805    1837    1876    1899    1942    1978    2010    2036    2054
          2074    2099    2119    2149    2174    2197    2220    2243    2277    2305    2345    2376    2415    2441    2470
          2497    2523    2539    2555    2581    2621    2647    2679    2707    2726    2746    2770    2796    2817    2842
          2863    2888    2905    2922    2933    2951    2969    2991    3025    3059    3082    3113    3136    3158    3186
          3210    3238    3271    3306    3334    3370    3406    3441    3480    3509    3537    3559    3579    3611    3653
          3701    3738    3764    3789    3809    3832    3851    3870    3892    3910    3934    3961    3985    4031    4109
          4218    4246    4267    4289    4310    4345    4378    4428    4442    4464    4497    4541    4583    4618    4659
          4684    4725    4777    4818    4858    4900    4947    5002    5027    5071    5109    5145    5174    5218    5239
          5254    5270    5285    5300    5315    5338    5356    5377    5398    5419    5440    5461    5490    5511    5533
          5562    5580    5608    5650    5669    5696    5745    5764    5782    5809    5845    5855    5865    5876    5886
          5900    5915    5927    5941    6035    6097    6145    6182    6195    6206    6232    6242    6253    6264    6274
          6284    6297    6308    6330    6342    6356    6371    6387    6408    6425    6445    6471    6537    6562    6583
          6607    6623    6646    6679    6698    6716    6736    6769    6782    9315    9334    9353    9381    9411    9423
          9435    9447    9459    9471    9517    9534    9563    9597    9635    9672    9708    9743    9826    9961   10065
         10139   10189   10240   10286   10351   10433   10529   10624   10661   10745   10776   10797   10834   10901   11038
         11180   11300   11331   11553   11589   11614   11702   11759   11787   11900   12004   12109   12506   12704   12805
         12870   12932   13005   13066   13132   13199   13251   13310   13455   13558   13668   13735   13801   13901   14077
         14249   14441   14516   14603   14776   15002   15207   15451   15549   15728   15883   16061   16182   16315   16460
         16598   16749   16974   17240   17400   17562   17729   17776   17836   17920   17982   18042   18095   18145   18191
```

```
          18242   18422   18597   18624   18671   18696   18723   18779   18830   18877   18928   18979   19031   19077   19128
          19171   19220   19268   19315   19363   19413   19464   19513   19560   19606   19658   19711   19979   20015   20052
          20083   20116   20148   20180   20217   20246   20285   20529   20702   21005   21044   21084   21122   21162   21202
          21243   21284   21312   21341   21601   21625   21756   21797   21827   21844   21970   21984   21997   22024   22040
          22053   22070   22092   22120   22149   22173   22198   22226   22252   22275   22291   22319   22348   22379   22403
          22426   22450   22492   22504   22535   22564   22608   22642   22668   22691   22762   22776   22789   22816   22837
          22850   22867   22895   22927   22960   22984   23009   23037   23068   23095   23109   23134   23160   23189   23211
          23238   23260   23288   23305   23363   23403
NXTTST     507#    9593    9629    9668    9704   10131   10492   10586
POP          1#
PUSH         1#
REPORT       1#
RSET     11525#   11581   11607   11694   11752   11779   11892   11996   12100   12498   12696   12798   12862   12924   12997
          13058   13124   13191   13243   13303   13550   13661   13729   13793   13895   14070   14243   14435   14510   14587
          14768   14994   15197   15441   15540   15719   15874   16052   16173   16305   16450   16588   16738   16964   17230
          17390   17554   17719   17766   17826   17913   17975   18035   18088   18138   18184   18235   18415   18591   18617
          18664   18689   18716   18772   18823   18871   18922   18973   19025   19071   19122   19165   19214   19262   19309
          19357   19406   19457   19507   19554   19600   19651   19705   19970   20007   20044   20077   20110   20142   20174
          20208   20238   20276   20521   20694   20996   21036   21075   21113   21153   21194   21235   21277   21305   21334
          21593   21617   21749   21789   21819   21897
SETPRI       1#
SETUP        1#
SKIP         1#
SLASH        1#
STARS        1#    507#     552     563     590     592     599     610     612     641     643     648     685     707     709
            758     760     787     789     817     820     833     835     862     864     892     895     899     901     934
            936     963     965     993    1013    1015    1023    1025    1033    1035    1043    1045    1052    1066    1068
           1077    1079    1090    1092    1105    1107    1116    1118    1131    1151    1153    1161    1163    1170    1172
           1179    1181    1188    1190    1203    1205    1218    1220    1233    1235    1249    1251    1264    1266    1280
           1282    1295    1297    1312    1314    1327    1329    1343    1345    1358    1360    1374    1388    1390    1398
           1400    1407    1409    1416    1418    1424    1439    1443    1455    1457    1472    1481    1483    1502    1508
           1510    1522    1530    1532    1546    1553    1555    1574    1583    1585    1605    1615    1617    1638    1648
           1650    1671    1676    1678    1702    1707    1709    1736    1738    1765    1767    1791    1805    1807    1825
           1837    1839    1863    1876    1878    1899    1901    1934    1942    1944    1963    1978    1980    2002    2010
           2012    2027    2036    2038    2054    2056    2074    2076    2099    2101    2119    2121    2141    2149    2151
           2166    2174    2176    2189    2197    2199    2212    2220    2222    2236    2243    2245    2270    2277    2279
           2297    2305    2307    2338    2345    2347    2367    2376    2378    2407    2415    2417    2434    2441    2443
           2463    2470    2472    2490    2497    2499    2517    2523    2525    2535    2539    2541    2551    2555    2557
           2573    2581    2583    2616    2621    2623    2641    2647    2649    2674    2679    2681    2700    2707    2709
           2720    2726    2728    2739    2746    2748    2759    2770    2772    2789    2796    2798    2810    2817    2819
           2833    2842    2844    2856    2863    2865    2880    2888    2890    2898    2905    2907    2915    2922    2924
           2933    2935    2951    2953    2969    2971    2991    2993    3025    3027    3059    3061    3082    3084    3113
           3115    3136    3138    3158    3160    3180    3186    3188    3204    3210    3212    3233    3238    3240    3266
           3271    3273    3299    3306    3308    3326    3334    3336    3361    3370    3372    3398    3406    3408    3434
           3441    3443    3468    3480    3482    3500    3509    3511    3527    3537    3539    3549    3559    3561    3571
           3579    3581    3602    3611    3613    3646    3653    3655    3694    3701    3703    3730    3738    3740    3754
           3764    3766    3782    3789    3791    3802    3809    3811    3825    3832    3834    3844    3851    3853    3863
           3870    3872    3885    3892    3894    3903    3910    3912    3925    3934    3936    3952    3961    3963    3975
           3985    3987    4000    4031    4033    4093    4109    4111    4211    4218    4220    4232    4246    4248    4267
           4269    4289    4291    4310    4312    4334    4345    4347    4378    4380    4419    4428    4430    4442    4444
           4464    4466    4487    4497    4499    4541    4543    4573    4583    4585    4618    4620    4659    4661    4674
           4684    4686    4725    4727    4767    4777    4779    4818    4820    4858    4860    4900    4902    4947    4949
           5002    5004    5027    5029    5063    5071    5073    5101    5109    5111    5138    5145    5147    5167    5174
           5176    5209    5218    5220    5239    5241    5254    5256    5270    5272    5285    5287    5300    5302    5315
           5317    5330    5338    5340    5356    5358    5377    5379    5398    5400    5419    5421    5440    5442    5461
           5463    5482    5490    5492    5506    5511    5513    5526    5533    5535    5557    5562    5564    5580    5582
```

	5601	5608	5610	5644	5650	5652	5663	5669	5671	5682	5696	5698	5735	5746	5748
	5764	5766	5782	5784	5809	5811	5838	5845	5847	5855	5857	5865	5867	5876	5878
	5886	5888	5895	5899	5900	5902	5913	5915	5917	5927	5929	5941	5943	5991	5995
	6015#	6036	6038	6097	6099	6145	6147	6182	6184	6195	6197	6206	6208	6232	6234
	6242	6244	6253	6255	6264	6266	6274	6276	6284	6286	6297	6299	6308	6310	6330
	6332	6342	6344	6356	6358	6371	6373	6387	6389	6408	6410	6425	6427	6445	6447
	6471	6473	6537	6539	6562	6564	6583	6585	6607	6609	6623	6625	6646	6648	6679
	6681	6698	6700	6716	6718	6736	6738	6769	6771	6782	6784	9315	9317	9334	9336
	9353	9355	9381	9383	9397	9408	9411	9413	9423	9425	9435	9437	9447	9449	9459
	9461	9471	9473	9517	9519	9534	9536	9563	9565	9597	9599	9635	9637	9672	9674
	9708	9710	9743	9745	9826	9828	9961	9963	10065	10067	10139	10141	10189	10191	10240
	10242	10274	10284	10286	10288	10351	10353	10414	10429	10433	10435	10529	10531	10624	10626
	10661	10663	10745	10747	10776	10778	10797	10799	10834	10836	10893	10898	10901	10903	11038
	11040	11180	11182	11300	11302	11331	11333	11361	11368	11380	11386	11396	11404	11425	11433
	11492	11507	11509	11553	11555	11589	11591	11614	11616	11702	11704	11759	11761	11787	11789
	11900	11902	12004	12006	12109	12111	12506	12508	12704	12706	12805	12807	12870	12872	12932
	12934	13005	13007	13066	13068	13132	13134	13199	13201	13251	13253	13310	13312	13455	13457
	13558	13560	13668	13670	13735	13737	13801	13803	13901	13903	14077	14079	14249	14251	14441
	14443	14516	14518	14603	14605	14776	14778	15002	15004	15207	15209	15451	15453	15549	15551
	15728	15730	15883	15885	16061	16063	16182	16184	16315	16317	16460	16462	16598	16600	16749
	16751	16974	16976	17240	17242	17400	17402	17562	17564	17729	17731	17776	17778	17836	17838
	17920	17922	17982	17984	18042	18044	18095	18097	18145	18147	18191	18193	18242	18244	18422
	18424	18597	18599	18624	18626	18671	18673	18696	18698	18723	18725	18779	18781	18830	18832
	18877	18879	18928	18930	18979	18981	19031	19033	19077	19079	19128	19130	19171	19173	19220
	19222	19268	19270	19315	19317	19363	19365	19413	19415	19464	19466	19513	19515	19560	19562
	19606	19608	19658	19660	19711	19713	19979	19981	20015	20017	20052	20054	20083	20085	20116
	20118	20148	20150	20180	20182	20217	20219	20246	20248	20285	20287	20529	20531	20702	20704
	21005	21007	21044	21046	21084	21086	21122	21124	21162	21164	21202	21204	21243	21245	21284
	21286	21312	21314	21341	21343	21601	21603	21625	21627	21756	21758	21797	21799	21827	21829
	21844	21846	21918	21970	21972	21984	21986	21997	21999	22024	22026	22040	22042	22053	22055
	22070	22072	22092	22094	22120	22122	22149	22151	22173	22175	22198	22200	22226	22228	22252
	22254	22275	22277	22291	22293	22319	22321	22348	22350	22379	22381	22403	22405	22426	22428
	22450	22452	22492	22494	22504	22506	22535	22537	22564	22566	22608	22610	22642	22644	22668
	22670	22691	22693	22762	22764	22776	22778	22789	22791	22816	22818	22837	22839	22850	22852
	22867	22869	22895	22897	22927	22929	22960	22962	22984	22986	23009	23011	23037	23039	23068
	23070	23095	23097	23109	23111	23134	23136	23160	23162	23189	23191	23211	23213	23238	23240
	23260	23262	23288	23290	23305	23307	23363	23365	23403	23405	23576	23579	23603	23606	
SWRSU	1#														
TIMMSG	9286#														
TIMTST	9286#	9427	9439	9451	9463										
TRAPTS	756#	758	787	833	862	899	934	963							
TYPBIN	1#														
TYPDEC	1#														
TYPNAM	1#														
TYPNUM	1#														
TYPOCS	1#														
TYPOCT	1#														
TYPTXT	1#														
USER	9286#	9287													
VTRP	6231#	6232	6242	6264	6274	6284	6297								
WMSG	9286#														
WTST	9286#	10143	10194												
$ASHC	6934#	7608	7628	7648	7670	7690	7710	7732	7752						
$ASHCS	6934#	7780	7803	7826	7849	7872	7895	7918	7941	7964	7987				
$ASHS	6934#	7206	7223	7240	7257	7274	7291	7308	7325	7342	7359				
$DIV	8017#	8632	8653	8674	8697	8718	8739	8762	8783	8804	8825	8848	8869	8890	8913

```
          8934   8955   8973   8993   9014   9035   9056   9077   9098   9119   9140   9161   9182
$MUL      8017#  8024   8045   8066   8087   8110   8131   8152   8175   8196   8217   8238   8261   8282   8303
          8324   8347   8368   8389   8416   8437   8458   8479   8500   8521   8542   8563   8584   8605
$SKIP      507#  5497   6614   6630   9752  10669  22543  22615  22649  22796  22823  22886  23044  23218  23267
         23370
$$ERCD     507#   740    768    782    797    811    843    857    872    886    909    923    944    958    973
           987   1021   1031   1041   1050   1075   1085   1088   1103   1114   1129   1159   1168   1177   1186
          1201   1217   1231   1247   1262   1278   1293   1309   1325   1341   1356   1372   1396   1405   1414
          1423   1461   1466   1469   1488   1500   1514   1520   1537   1544   1562   1571   1593   1603   1626
          1636   1659   1669   1688   1700   1719   1723   1732   1735   1749   1753   1761   1764   1778   1786
          1789   1815   1823   1849   1861   1887   1898   1916   1919   1924   1927   1932   1951   1961   1992
          2000   2019   2025   2046   2052   2067   2070   2073   2091   2094   2098   2115   2118   2133   2139
          2156   2163   2187   2210   2234   2257   2268   2291   2295   2319   2322   2332   2336   2361   2365
          2393   2402   2406   2429   2432   2457   2461   2485   2488   2512   2515   2533   2549   2568   2571
          2588   2595   2600   2609   2614   2636   2640   2656   2659   2663   2669   2673   2693   2699   2718
          2737   2757   2783   2787   2808   2827   2831   2852   2855   2875   2879   2897   2914   2931   2946
          2949   2964   2967   2981   2986   2989   3005   3008   3015   3019   3022   3042   3045   3053   3057
          3074   3077   3080   3097   3100   3105   3108   3111   3128   3131   3134   3150   3153   3156   3172
          3175   3178   3199   3202   3223   3228   3231   3249   3252   3258   3261   3264   3283   3286   3289
          3293   3297   3318   3321   3324   3344   3347   3353   3356   3359   3382   3385   3388   3393   3396
          3418   3421   3424   3429   3432   3453   3456   3459   3463   3466   3492   3521   3548   3570   3591
          3600   3624   3633   3644   3668   3679   3692   3712   3720   3728   3752   3779   3800   3821   3839
          3842   3860   3879   3882   3900   3919   3922   3943   3947   3971   3995   4040   4043   4051   4057
          4060   4067   4074   4082   4090   4125   4138   4151   4163   4175   4185   4195   4208   4228   4231
          4257   4266   4279   4288   4301   4309   4323   4332   4357   4367   4376   4390   4398   4406   4416
          4440   4454   [illegible] 4476 4485 4508   4516   4524   4530   4539   4553   4562   4571   4596   4606
          4616   4630   [illegible] 4649 4657 4671   4696   4705   4714   4723   4738   4747   4756   4765   4790
          4799   4807   [illegible] 4831 4840 4849   4857   4871   4880   4888   4899   4913   4923   4932   4944
          4957   4964   4970   4973   4986   4989   4996   4999   5014   5017   5024   5037   5044   5049   5058
          5061   5086   5100   5125   5137   5156   5166   5185   5194   5201   5207   5227   5237   5249   5252
          5265   5268   5280   5283   5295   5298   5310   5313   5325   5328   5346   5354   5369   5372   5375
          5390   5393   5396   5411   5414   5417   5432   5435   5438   5453   5456   5459   5474   5477   5480
          5503   5522   5542   5546   5552   5573   5577   5593   5597   5641   5660   5681   5711   5728   5756
          5762   5774   5780   5794   5798   5803   5807   5821   5826   5831   5835   5853   5863   5874   5884
          5894   5909   5925   5939   5954   5998   6000   6002   6004   6006   6008   6010   6012   6014   6044
          6050   6056   6063   6070   6077   6083   6090   6096   6108   6117   6126   6135   6144   6152   6156
          6160   6164   6167   6171   6175   6179   6192   6204   6219   6224   6229   6296   6327   6340   6352
          6355   6368   6384   6401   6406   6419   6423   6440   6442   6458   6486   6491   6494   6512   6519
          6523   6557   6577   6579   6599   6621   6640   6669   6672   6676   6690   6693   6709   6723   6726
          6746   6748   6751   6754   6757   6836   6839   6843   6972   6977   7001   7006   7029   7034   7057
          7062   7086   7091   7116   7121   7200   7215   7218   7232   7235   7249   7252   7266   7269   7283
          7286   7300   7303   7317   7320   7334   7337   7351   7354   7368   7371   7409   7413   7418   7422
          7446   7450   7455   7458   7483   7487   7492   7495   7604   7620   7623   7640   7643   7661   7664
          7682   7685   7702   7705   7723   7726   7744   7747   7764   7767   7770   7792   7795   7798   7815
          7818   7821   7838   7841   7844   7861   7864   7867   7884   7887   7890   7907   7910   7913   7930
          7933   7936   7953   7956   7959   7976   7979   7982   7999   8002   8005   8034   8037   8040   8055
          8058   8061   8076   8079   8082   8098   8101   8104   8120   8123   8126   8141   8144   8147   8163
          8166   8169   8185   8188   8191   8206   8209   8212   8227   8230   8233   8249   8252   8255   8271
          8274   8277   8292   8295   8298   8313   8316   8319   8335   8338   8341   8357   8360   8363   8378
          8381   8384   8400   8403   8406   8426   8429   8432   8447   8450   8453   8468   8471   8474   8489
          8492   8495   8510   8513   8516   8531   8534   8537   8552   8555   8558   8573   8576   8579   8594
          8597   8600   8615   8618   8621   8643   8646   8649   8664   8667   8670   8686   8689   8692   8708
          8711   8714   8729   8732   8735   8751   8754   8757   8773   8776   8779   8794   8797   8800   8815
          8818   8821   8837   8840   8843   8859   8862   8865   8880   8883   8886   8902   8905   8908   8924
          8927   8930   8945   8948   8951   8968   8985   9004   9007   9010   9025   9028   9031   9046   9049
          9052   9067   9070   9073   9088   9091   9094   9109   9112   9115   9130   9133   9136   9151   9154
```

	9157	9172	9175	9178	9193	9196	9199	9212	9215	9326	9345	9365	9376	9392	9481
	9489	9500	9509	9527	9541	9548	9557	9573	9583	9607	9619	9649	9661	9686	9697
	9761	9773	9784	9797	9859	9883	9907	9932	9953	9983	10007	10031	10055	10101	10133
	10161	10171	10180	10211	10221	10230	10253	10265	10309	10316	10324	10331	10367	10374	10382
	10389	10460	10470	10477	10485	10554	10564	10571	10579	10644	10678	10686	10709	10726	10735
	10763	10791	10807	10817	10824	10881	10946	10958	10969	10979	10990	11004	11016	11031	11056
	11073	11089	11103	11105	11120	11137	11153	11172	11210	11222	11233	11243	11254	11267	11279
	11293	11314	11324	11344	11351	11442	11456	11469	11483	11577	11605	11627	11637	11647	11656
	11666	11675	11684	11693	11721	11735	11740	11751	11778	11813	11816	11822	11828	11846	11849
	11855	11862	11931	11958	11972	12034	12061	12075	12462	12649	12733	12736	12759	12761	12766
	12782	12827	12833	12838	12890	12896	12901	12964	12973	13028	13035	13039	13091	13098	13159
	13165	13222	13226	13275	13280	13339	13357	13372	13387	13409	13414	13420	13473	13477	13494
	13498	13513	13517	13534	13538	13576	13581	13596	13601	13616	13621	13636	13641	13690	13694
	13756	13760	13819	13838	13860	13919	13923	13938	13942	13958	13962	13978	13982	13997	14001
	14016	14020	14095	14100	14115	14119	14136	14140	14156	14160	14175	14179	14194	14198	14267
	14272	14287	14292	14307	14312	14327	14331	14346	14350	14365	14370	14385	14390	14459	14464
	14479	14484	14534	14538	14554	14558	14623	14627	14647	14652	14670	14675	14690	14695	14710
	14715	14796	14801	14813	14817	14826	14831	14837	14854	14859	14873	14882	14887	14893	14910
	14915	14927	14931	14940	14945	14951	15015	15024	15029	15042	15053	15059	15072	15086	15104
	15128	15134	15149	15155	15424	15431	15474	15478	15495	15499	15513	15519	15527	15531	15707
	15710	15713	15864	15869	16040	16043	16046	16163	16168	16294	16299	16439	16444	16556	16560
	16574	16579	16582	16707	16711	16725	16729	16732	16939	16942	16946	16949	16952	17209	17217
	17221	17254	17277	17299	17361	17364	17368	17371	17374	17419	17448	17526	17534	17538	17570
	17577	17582	17586	17592	17596	17603	17607	17612	17617	17621	17626	17632	17636	17641	17648
	17655	17664	17672	17680	17689	17697	17745	17751	17761	17764	17801	17825	17862	17884	17902
	17912	17959	18034	18086	18137	18183	18234	18412	18588	18615	18652	18688	18714	18755	18822
	18870	18921	18972	19024	19070	19121	19164	19213	19261	19308	19356	19405	19456	19506	19553
	19599	19650	19703	19964	20005	20043	20076	20109	20141	20173	20207	20236	20263	20267	20515
	20692	20990	21034	21073	21112	21152	21193	21234	21276	21298	21327	21587	21735	21738	21742
	21781	21810	21813	21866	21882	21979	21993	22014	22020	22036	22049	22062	22076	22080	22084
	22089	22102	22106	22110	22116	22130	22134	22138	22144	22161	22166	22187	22194	22213	22222
	22244	22264	22271	22287	22308	22312	22334	22343	22364	22374	22398	22422	22440	22443	22465
	22501	22514	22518	22522	22532	22547	22554	22560	22580	22589	22597	22628	22636	22664	22685
	22725	22771	22785	22806	22812	22833	22846	22859	22873	22877	22881	22892	22905	22909	22913
	22923	22937	22941	22945	22955	22972	22977	22998	23005	23024	23033	23060	23078	23088	23105
	23123	23127	23148	23155	23174	23184	23206	23234	23250	23253	23279	23283	23299	23302	23317
	23321	23393	23463	23480	23498	23504	23509	23515							
$$ERRO	507#	1020	1030	1040	1049	1074	1087	1101	1113	1127	1158	1167	1176	1185	1199
	1215	1229	1245	1260	1276	1291	1307	1323	1339	1354	1370	1395	1404	1413	1422
	1468	1498	1518	1542	1569	1601	1634	1667	1698	1734	1763	1788	1821	1859	1896
	1931	1959	1998	2023	2050	2072	2097	2117	2138	2161	2185	2208	2232	2266	2294
	2335	2364	2405	2431	2460	2487	2514	2532	2548	2570	2613	2639	2672	2697	2717
	2736	2756	2786	2807	2830	2854	2878	2896	2913	2930	2948	2966	2988	3021	3056
	3079	3110	3133	3155	3177	3201	3230	3263	3296	3323	3358	3395	3431	3465	3490
	3519	3547	3569	3598	3642	3690	3726	3750	3777	3798	3819	3841	3859	3881	3899
	3921	3945	3969	3993	4089	4206	4230	4264	4286	4307	4330	4374	4414	4438	4461
	4483	4537	4569	4614	4655	4669	4721	4763	4815	4855	4897	4942	4998	5022	5060
	5098	5135	5164	5205	5235	5251	5267	5282	5297	5312	5327	5353	5374	5395	5416
	5437	5458	5479	5502	5596	5679	5761	5779	5806	5834	5852	5862	5873	5883	5893
	5908	5924	5938	6095	6143	6178	6203	6295	6354	6367	6383	6422	6620	6692	6708
	6725	9375	9391	9556	9952	10308	10459	10553	10790	11102	17750	22035	22088	22286	22832
	22891	23104													
$$ESCA	1#														
$$NEWT	1#	507#	707	758	787	833	862	899	934	963	1013	1023	1033	1043	1066
	1077	1090	1105	1116	1151	1161	1170	1179	1188	1203	1218	1233	1249	1264	1280
	1295	1312	1327	1343	1358	1388	1398	1407	1416	1455	1481	1508	1530	1553	1583

```
          1615   1648   1676   1707   1736   1765   1805   1837   1876   1899   1942   1978   2010   2036   2054
          2074   2099   2119   2149   2174   2197   2220   2243   2277   2305   2345   2376   2415   2441   2470
          2497   2523   2539   2555   2581   2621   2647   2679   2707   2726   2746   2770   2796   2817   2842
          2863   2888   2905   2922   2933   2951   2969   2991   3025   3059   3082   3113   3136   3158   3186
          3210   3238   3271   3306   3334   3370   3406   3441   3480   3509   3537   3559   3579   3611   3653
          3701   3738   3764   3789   3809   3832   3851   3870   3892   3910   3934   3961   3985   4031   4109
          4218   4246   4267   4289   4310   4345   4378   4428   4442   4464   4497   4541   4583   4618   4659
          4684   4725   4777   4818   4858   4900   4947   5002   5027   5071   5109   5145   5174   5218   5239
          5254   5270   5285   5300   5315   5338   5356   5377   5398   5419   5440   5461   5490   5511   5533
          5562   5580   5608   5650   5669   5696   5746   5764   5782   5809   5845   5855   5865   5876   5886
          5900   5915   5927   5941   6036   6097   6145   6182   6195   6206   6232   6242   6253   6264   6274
          6284   6297   6308   6330   6342   6356   6371   6387   6408   6425   6445   6471   6537   6562   6583
          6607   6623   6646   6679   6698   6716   6736   6769   6782   9315   9334   9353   9381   9411   9423
          9435   9447   9459   9471   9517   9534   9563   9597   9635   9672   9708   9743   9826   9961  10065
         10139  10189  10240  10286  10351  10433  10529  10624  10661  10745  10776  10797  10834  10901  11038
         11180  11300  11331  11553  11589  11614  11702  11759  11787  11900  12004  12109  12506  12704  12805
         12870  12932  13005  13066  13132  13199  13251  13310  13455  13558  13668  13735  13801  13901  14077
         14249  14441  14516  14603  14776  15002  15207  15451  15549  15728  15883  16061  16182  16315  16460
         16598  16749  16974  17240  17400  17562  17729  17776  17836  17920  17982  18042  18095  18145  18191
         18242  18422  18597  18624  18671  18696  18723  18779  18830  18877  18928  18979  19031  19077  19128
         19171  19220  19268  19315  19363  19413  19464  19513  19560  19606  19658  19711  19979  20015  20052
         20083  20116  20148  20180  20217  20246  20285  20529  20702  21005  21044  21084  21122  21162  21202
         21243  21284  21312  21341  21601  21625  21756  21797  21827  21844  21970  21984  21997  22024  22040
         22053  22070  22092  22120  22149  22173  22198  22226  22252  22275  22291  22319  22348  22379  22403
         22426  22450  22492  22504  22535  22564  22608  22642  22668  22691  22762  22776  22789  22816  22837
         22850  22867  22895  22927  22960  22984  23009  23037  23068  23095  23109  23134  23160  23189  23211
         23238  23260  23288  23305  23363  23403
$$SKIP       1#
.EQUAT       1#
.HEADE       1#
.KT11        1#   9217#   9218
.MSG     21826#
.SETUP       1#    550#   6929
.SWRHI       1#
.$ACT1       1#    550#
.$APTB       1#    550#    561
.$APTH       1#    550#    588
.$APTY       1#
.$ASTA       1#
.$CATC       1#
.$CMTA       1#
.$DB2D       1#
.$DB2O       1#
.$DIV        1#
.$EOP        1#
.$ERRO       1#
.$ERRT       1#
.$MULT       1#
.$POWE       1#
.$RAND       1#
.$RDDE       1#
.$RDOC       1#
.$READ       1#
.$R2AZ       1#
.$SAVE       1#
.$SB2D       1#
```

```
.$SB20      1#
.$SCOP      1#
.$SIZE      1#
.$SUPR      1#
.$TRAP      1#
.$TYPB      1#
.$TYPD      1#
.$TYPE      1#
.$TYPO      1#
.$40CA      1#
.1170       1#

. ABS.  131741     000

 ERRORS DETECTED:  0

 CJKDEB.BIN,CJKDEB.LST/CRF/NL:TOC/SOL=SYSMAC.SML,CJKDEB.P11
 RUN-TIME: 63 75 11 SECONDS
 RUN-TIME RATIO: 432/150=2.8
 CORE USED:  51K  (102 PAGES)
```